AFRL-AFOSR-UK-TR-2015-0015

AMULET: A MUlti-cLuE Approach to Image Forensics

Dr. Mauro Barni

UNIVERSITA DEGLI STUDI DI SIENA
LOCALITA BANCHI DI SOTTO 55
SIENA, ITALY

EOARD GRANT #FA8655-12-1-2138

Report Date: December 2014

Final Report from 1 October 2012 to 31 December 2014

Air Force Research Laboratory
Air Force Office of Scientific Research
European Office of Aerospace Research and Development
Unit 4515, APO AE 09421-4515

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
31 December 2014	Final Report	1 October 2012 – 31 December 2014

4. TITLE AND SUBTITLE

AMULET: A MUlti-cLuE Approach to Image Forensics

5a. CONTRACT NUMBER

5b. GRANT NUMBER

FA8655-12-1-2138

5c. PROGRAM ELEMENT NUMBER

61102F

6. AUTHOR(S)

Dr. Mauro Barni

5d. PROJECT NUMBER

5d. TASK NUMBER

5e. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

UNIVERSITA DEGLI STUDI DI SIENA
LOCALITA BANCHI DI SOTTO 55
SIENA, ITALY

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

EOARD
Unit 4515
APO AE 09421-4515

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/AFOSR/IOE (EOARD)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

AFRL-AFOSR-UK-TR-2015-0015

12. DISTRIBUTION/AVAILABILITY STATEMENT

Distribution A: Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The goal of AMULET (A MUlti-cLuE approach To image forensics) was the development of multi-clue forensics techniques that, starting from the indications provided by a pool of forensic tools focusing on specific artifacts, reach a global conclusion about the authenticity of an image. The techniques had to operate in highly non-structured scenarios characterized by imprecise and incomplete information. The main output of the research activity has been the development of a non-conventional multi-clue inference framework based on Dempster-Shafe (DST) Theory of evidence.

15. SUBJECT TERMS

EOARD, Nano particles, Photo-Acoustic Sensors

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18, NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			James H Lawton, PhD
UNCLAS	UNCLAS	UNCLAS	SAR	72	19b. TELEPHONE NUMBER *(Include area code)* (703)696-5999

AMULET:
A MUlti-cLuE approach To image forensics
Final Activity Report

Mauro Barni, Marco Fontani, Benedetta Tondi
Department of Information Engineering and Mathematics
University of Siena

Abstract

The goal of AMULET was the development of multi-clue forensics techniques that, starting from the indications provided by a pool of forensic tools focusing on specific artifacts, reach a global conclusion about the authenticity of an image. The techniques had to operate in highly non-structured scenarios characterized by imprecise and incomplete information.

The main output of the research activity has been the development of a non-conventional multi-clue inference framework based on Dempster-Shafe (DST) Theory of evidence.

As opposed to other widely adopted approaches based on machine learning, like Support Vector Machines (SVM) or Neural Networks (NN), the proposed framework permits to obtain deeper insights about the achievable performance and the reasons behind the performance of the system. The developed system focuses on local image tampering and considers both supervised and unsupervised image authentication. In the former case, the system is able to decide if a region indicated by the user is authentic or not, in the latter case, the possibly tampered region is automatically localized by the system, though at the price of a reduced reliability. In both cases, the decision fusion framework does not rely only on the information provided by the forensics tools, but also on the additional information coming from the practical conditions under which the tools must operate (e.g. size of the analyzed image, image quality). The possibility of using the multi-clue framework to counter anti-forensic techniques has also been investigated, by including into the decision process the information provided by tools explicitly designed to detected the traces left by anti-forensic tools. The performance and flexibility of the DST fusion framework have been compared with a number of alternative approaches, including: Bayesian inference, machine learning (SVM) and fuzzy logic. The performance of the proposed system are better than those of the competing approaches, in addition, system tuning is considerably easier due to the reduced number of to-be-set parameters and the absence of heavy training procedures.

In parallel to the development of the DS fusion framework, AMULET focused on the Information Theoretic analysis of multi-clue forensics, with particular attention to the adversarial version of the problem, that is when an adversary deliberately tries to undermine the performance of the system. This research line led to the definition of a theoretical background for multiclue-based forensics, which thanks to a proper game-theoretic formulation, permits to evaluate the ultimately achievable performance when an adversary counters the analysis.

Contents

1 Introduction

During the two years of the project, the activity of AMULET focused on two main research lines: i) the development of a multi-clue approach that starting from the evidence provided by different analysis tools decide upon the authenticity of a given image, and ii) the development of a coherent and rigorous framework to analyze the ultimate performance achievable when the action of the forensic analyst is countered by an adversary explicitly aiming at system failure.

A great deal of the research was devoted to the former line. The activity, concentrated on the the usage of Dempster-Shafer Theory (DST) of evidence for taking a decision on image authenticity starting form several heterogenous, incomplete and sometimes unreliable clues. Such a choice was motivated by the observation that with respect to more classical approaches to inference reasoning, the use of DST avoids the necessity of knowing a-priori probabilities (that would be extremely difficult to estimate in an multimedia forensics scenario) and also provides more intuitive means for managing the uncertainty characterizing the information provided by the forensic tools. Among the advantages deriving from the adoption of DST we mention i) the possibility of exploiting all available information about tools reliability and about the compatibility between the traces the forensic tools look for, and ii) the ease with which the framework can be extended by incrementally adding new tools with a little effort. The possibility of using the proposed multi-clue approach to counter anti-forensic techniques was also considered addressed. In particular, we adapted the DST fusion framework to allow the incorporation of the information provided by tools explicitly looking for the traces left by anti-forensic techniques. The multi-clue DST framework was applied to two different scenarios. In a first case, the system is asked to judge the authenticity of an image region selected by the user (tampering detection). In a second case, the system must localize the presence of a tampered region without any help from the user (tampering localization). In both cases, the performance of the proposed approach outperformed (in some cases only slightly) the performance of competing systems, this proving the validity of the adopted solutions.

From a more theoretical point of view, part of the activity was devoted to analyzing the ultimate achievable performance of any forensics scheme operating in an adversarial set-up, i.e. in the presence of one or more adversaries with the explicitly aim of making the analysis fail. This activity is part of a larger research effort, aiming at building a rigorous theoretical background for multimedia forensics and, more in general, to develop a coherent theory of adversarial signal processing [1]. More specifically, and by considering the particular goal of AMULET project, we extended the analysis carried out in [2], to address the problem of binary hypothesis testing based on multiple observations in the presence of an adversary corrupting part or all the observations. Despite its theoretical nature, the results we obtained provide a reference background that will turn out to be extremely useful to guide the future, and possibly more applied, research in the field.

Several publications originated from AMULET. They are listed below together with their abstract. The list includes also some publications that are not directly related to multi-clue forensics analysis, since they have been obtained as additional side-results of the theoretical part of the project.

List of publications

The DST fusion framework for tampering detection is described in the following papers.

- **M. Fontani, T. Bianchi, A. De Rosa, and M. Barni A. Piva, "A framework for decision fusion in image forensics based on Dempster-Shafer theory of evidence", *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 4, pp. 593-607, April 2013.**

  Abstract. In this work, we present a decision fusion strategy for image forensics. We define a framework that exploits information provided by available forensic tools to yield a global judgment about the authenticity of an image. Sources of information are modeled and fused using Dempster-Shafer Theory of Evidence, since this theory allows us to handle uncertain

answers from tools and lack of knowledge about prior probabilities better than the classical Bayesian approach. The proposed framework permits us to exploit any available information about tools reliability and about the compatibility between the traces the forensic tools look for. The framework is easily extendable: new tools can be added incrementally with a little effort. Comparison with logical disjunction- and SVM-based fusion approaches shows an improvement in classification accuracy, particularly when strong generalization capabilities are needed.

- **M. Fontani, T. Bianchi, A. De Rosa, A. Piva, and M. Barni, "A forensic tool for investigating image forgeries", *Intl. Journal of Digital Crime and Forensics (IJDCF)*, vol. 5, no. 4, pp. 15-33, 2013.**

  Abstract. Images have always been considered a reliable source of evidence in the past. Today, the wide availability of photo editing software urges us to investigate the origin and the integrity of a digital image before trusting it. Although several algorithms have been developed for image integrity verification, a comprehensive tool that allows the analyst to synergically exploit these algorithms, and to reach a final decision based on their output, is still lacking. In this work we propose an image forensic tool trying to fill this gap. The proposed tool exploits state of the art algorithms for splicing detection, with forgery localization capabilities, and make them available to the analyst through a graphical interface. In order to help the analyst in reaching a final assessment, a decision fusion engine is employed to intelligently merge the output of different algorithms, boosting detection performance. The tool has a modular architecture, that makes it easily scalable.

- **M. Fontani, A. Aragones-Rua, C. Troncoso, and M. Barni, The watchful forensic analyst: Multi-clue information fusion with background knowledge, *in WIFS 2013, IEEE Intl. Workshop on Information Forensics and Security*, Guangzhou, China, 18-21 November 2013.**

  Abstract. Image Forensics (IF) is a challenging research topic, that suffers from strong limitations when facing with real world applications. A possible way to cope with these limitations is to resort to data fusion, whereby the outputs of different forensic tools are used to reach a final decision about the analyzed image. Nevertheless, existing schemes do not take full advantage of all the information available to the analyst, like the knowledge of the dependence of the performance of forensic tools on side conditions. Specifically, in this paper we show how the performance of forensic tools varies according to a number of parameters, most of which are directly observable by the analyst. After showing some practical examples, we propose a method to cast this background information into two multi-clue information fusion frameworks, yielding a significant improvement of the overall performance at virtually no cost.

The incorporation of counter-anti-forensics tools within the DST inference framework is described in the following paper.

- **M. Fontani, A. Bonchi, A. Piva, M. Barni, Countering anti-forensics by means of data fusion, *in SPIE Conf. on Media Watermarking, Security, and Forensics*, S. Francisco (CA), USA, 3-5 February 2014.**

  Abstract. In the last years many image forensic (IF) algorithms have been proposed to reveal traces of processing or tampering. On the other hand, Anti-Forensic (AF) tools have also been developed to help the forger in removing editing footprints. Inspired by the fact that it is much harder to commit a perfect crime when the forensic analyst uses a multi-clue investigation strategy, we analyse the possibility offered by the adoption of a data fusion framework in a Counter-Anti-Forensic (CAF) scenario. We do so by adopting a theoretical framework, based on Dempster-Shafer Theory of Evidence, to synergically merge information provided by IF tools and CAF tools, whose goal is to reveal traces introduced by anti-forensic algorithms. The proposed system accounts for the non-trivial relationships between IF and CAF techniques; for example, in some cases the outputs from the former are expected to contradict the output from the latter. We evaluate the proposed method within a representative forensic task, that is splicing detection in JPEG images, with the forger trying to conceal traces using two different counter-forensic methods. Results show that decision fusion strongly limits the effectiveness of AF methods.

The DST multi-clue system for forgery localization has not been published yet. The following paper is under preparation.

- **P. Ferrara, M. Fontani, T. Bianchi, A. De Rosa, A. Piva, M. Barni, "Unsupervised Image Forgery Localization based on Dempster-Shafer Multi-Clue Analysis", *in preparation***

  Abstract. Image authenticity verification has usually to be carried out without any knowledge about the processing undergone by the image or the region that suffered some forgery. In this setting, it is fundamental to rely on a multi-clue analysis, that cleverly merges the information stemming from several complementary tools. This work introduces a fully automatic framework for fusing the maps output by a set of unsupervised forgery localization algorithms. The framework takes into account the forgery maps, their reliability and the compatibility among the different traces considered by the different tools. The achieved localization map is then refined by exploiting image content, thus improving the overall performance of the proposed system with respect to state of the art approaches.

The ultimate limits of multi-clue forensics analysis under adversarial conditions are studied in the following paper.

- **M. Barni and B. Tondi, Multiple-observation hypothesis testing under adversarial conditions, *in WIFS 2013, IEEE Intern. Workshop on Information Forensics and Security*, Guangzhou, China, 18-21 November 2013.**

  Abstract. We address the problem of binary hypothesis testing based on multiple observations in the presence of an adversary corrupting part or all the observations. We propose a general framework based on game-theory that encompasses a wide variety of situations including distributed detection, data fusion, multimedia forensics, sensor networks. The proposed approach extends the Neyman-Pearson approach to an adversarial setting in which the analyst must ensure that type I error probability stays below a threshold, and the adversary tries to induce a type II error. We derive the equilibrium point of the game in an asymptotic set up, showing that a dominant strategy exists for the analyst. The paper opens the way to further analysis in which the payoff of the game at the equilibrium is analyzed thus permitting to understand the ultimate achievable performance of multiple-observation hypothesis testing under adversarial conditions.

Other papers which originated from AMULET activity but which are not directly related to multi-clue forensics analysis are listed below.

- **M. Barni, M. Fontani, and B. Tondi, "Universal Counter-forensics of Multiple Compressed JPEG Images", in *IWDW 2014, IEEE Intl. Workshop on Digital Forensics and Watermarking*, Taipei, October 2014.**

  Abstract. Detection of multiple JPEG compression of digital images has been attracting more and more interest in the field of multimedia forensics. On the other side, techniques to conceal the traces of multiple compression are being proposed as well. Motivated by a recent trend towards the adoption of universal approaches, we propose a counter- forensic technique that makes multiple compression undetectable for any forensic detector based on the analysis of the histograms of quantized DCT coefficients. Experimental results show the effectiveness of our approach in removing the artifacts of double and also triple compression, while maintaining a good quality of the image.

- **B. Tondi, M. Barni, "Source Distinguishability under Corrupted Training", in *WIFS 2104, IEEE Intl. Workshop on Information Forensics and Security*, 3-5 December 2014, Atlanta, Georgia, USA.**

  Abstract. We study a new variant of the source identification game with training data in which part of the training data is corrupted by an adversary. In such a scenario, the defender wants to decide whether a test sequence xn has been drawn from the same source which generated a training sequence tN, part of which has been corrupted by the adversary. By adopting a game theoretical formulation, we derive the unique rationalizable equilibrium of the game in

the asymptotic setup. Moreover, by mimicking SteinÕs lemma, we derive the best achievable performance for the defender, permitting us to analyze the ultimate distinguishability of the two sources. We conclude the paper by comparing the performance of the test with corrupted training to the simpler case in which the adversary can not modify the training sequence, and by deriving the percentage of samples that the adversary needs to modify to make source identification impossible.

- **M. Barni, B. Tondi, "Source Distinguishability under Distortion-Limited Attack: an Optimal Transport Perspective", submitted for possible publication on IEEE Trans. on Information Theory.**

  Abstract. We analyze the distinguishability of two sources in a Neyman-Pearson set-up when an attacker is allowed to modify the output of one of the two sources subject to a distortion constraint. By casting the problem in a game-theoretic framework and by exploiting the parallelism between the attackerÕs goal and Optimal Transport Theory, we introduce the concept of Security Margin defined as the maximum average per-sample distortion introduced by the attacker for which the two sources can be distinguished ensuring arbitrarily small, yet positive, error exponents for type I and type II error probabilities. Several versions of the problem are considered according to the available knowledge about the sources and the type of distance used to define the distortion constraint. We compute the security margin for some classes of sources and derive a general upper bound assuming that the distortion is measured in terms of the mean square error between the original and the attacked sequence.

This report is organized as follows. In Section 2, we present the decision fusion framework that we have developed by relying on Dempster-Shafer theory of evidence. In Section 3, we describe the experiments that we carried out to validate the decision fusion framework. We do so, by considering a specific scenario in which double JOEG artifacts are sued for cut and past detection. Section 4 extends the analysis of the previous sections to the tampering localization problem. The next section (Section 5) illustrates the possible use of the proposed DST fusion framework to combat counter forensics techniques. Finally, in Section 6, we introduce a theoretical framework to analyze the ultimate achievable performance of multi-clue forensics analysis in an adversarial setup. We do so for the particular case of binary hypothesis testing in the presence of an attacker aiming at increasing the false negative error rate of the test.

2 Multi-clue forgery detection based on DST

In this section, we describe the multi-clue inference framework that we developed to merge the information provided by different forensic tools. Even if within AMULET we instantiated the framework to cope with a well defined forensic goal, namely deciding if a region indicated by the user has been copy-pasted from another image by relying on the traces left by double JPEG compression, our goal was a more general one, since we aimed at the definion of a theoretical model that allows fusing a generic set of forensic tools, requiring as few prior information as possible. To do so we relied on Dempster-Shafer Theory of Evidence (DST), since this theory can model uncertainty and missing information in a very intuitive and sound way, and does not require the knowledge of prior probabilities in the modeling phase.

In addition to the output of the available forensic tools, our framework exploits the knowledge about the reliability of the tools and the compatibility between different tampering traces. In addition it can be easily extended when new tools become available. It allows both a "soft" and a binary (tampered/non-tampered) interpretation of the fusion result, and can help in analyzing images for which taking a decision is critical due to conflicting data.

We tested the effectiveness of the proposed framework, by applying it to the splicing detection problem, which consists in determining whether a given region of an image has been pasted from another. During this process some traces are left into the image, depending on the modality used to create the forgery: the presence of these traces can be revealed by using one or more forensic tools, each of which provides information about the presence of the trace it is looking for. Note that, in splicing *detection*, most forensic tools assume knowledge of the suspect region.

This section is divided in two parts: the first one gives a brief introduction to DS theory, while the second presents the proposed framework. The experimental validation of the framework is presented in the next section.

2.1 Introduction to Desmpter-Shafer Theory of Evidence

Dempster-Shafer Theory of Evidence (DST) [3, 4] is a widely employed mathematical framework allowing to make reasoning and inference in contexts where uncertainty and lack of information are strong. Indeed, one of the most attractive features of DST is the capability of modeling uncertainty and doubt in an explicit and very simple way, especially compared to classical probability theory. When using classical probability theory for defining the probability of a certain event A, the additivity rule must be satisfied; so by saying that $Pr(A) = p_A$ one is also implicitly saying that $Pr(\bar{A}) = 1 - p_A$, thus committing the probability of an event A to that of its complementary $\bar{A}$. More importantly, the additivity rule influences also the representation of ignorance: complete ignorance about a dichotomic event A in Bayesian theory is commonly represented by setting $Pr(A) = Pr(\bar{A}) = 0.5$ (according to the maximum entropy principle), but this probability distribution also models perfect knowledge about the probability of each event being 0.5 (as for coin tossing), thus making it difficult to distinguish between ignorance and perfect knowledge about equiprobable events. To avoid such a problem, DS theory abandons the classical probability framework and allows to reason without the need to introduce a-priori probabilities.

2.1.1 Shafer's formalism

Let the frame $\Theta = \{x_1, x_2, \ldots, x_n\}$ define a finite set of possible values of a variable X; a proposition about X is any subset of Θ. We are interested in quantifying how much we are confident in propositions of the form "the true value of X is in H", where $H \subseteq \Theta$ (notice that the set of all possible propositions is the power set of Θ, 2^{Θ}). To give an example, let us think of a patient that can either be affected by cancer or not: we can model this scenario defining a variable C with frame $\Theta = \{ac, nc\}$ where ac is the proposition "the patient is affected by cancer", nc is the proposition "the patient is not affected by cancer", and $(ac \cup nc)$ is the doubtful proposition "the patient is or is not affected by cancer". Properly choosing the set Θ is very important in DST: it must represent the desired granularity of information that we want to (or we can) reach,

so that choosing $\Theta = \{\text{car}, \text{truck}, \text{bus}, \text{scooter}, \text{motorbike}\}$ may be less appropriate than choosing $\Theta = \{\text{four-wheeled}, \text{two-wheeled}\}$ for some tasks.

The link between propositions and subsets of Θ allows to map logical operations on propositions into operations between sets. Each proposition is mapped into a single subset and is assigned a basic belief *mass* through a Basic Belief Assignment, defined over the frame of the variable.

Definition 1. *Let Θ be a frame. A function $m^\Theta : 2^\Theta \to [0, 1]$ is called a Basic Belief Assignment (BBA) over the frame Θ if:*

$$m^\Theta(\emptyset) = 0; \qquad \sum_{A \in 2^\Theta} m^\Theta(A) = 1 \tag{1}$$

where the summation is taken over all possible subsets of Θ.

Continuing the previous example, after examining the patient a doctor could provide information that lead us to write the following basic belief assignment:

$$m^\Theta(X) = \begin{cases} 0.8 & \text{for } X = \{ac\} \\ 0.2 & \text{for } X = \{nc\} \\ 0 & \text{for } X = \{ac \cup nc\} \end{cases}. \tag{2}$$

Each set S such that $m^\Theta(S) > 0$ is called a *focal element* for m. In the following, we will omit the frame when it is clear from the context, writing m instead of m^Θ; furthermore, when writing mass assignments only focal elements will be listed (so the last row of eq. (2) would not appear). BBAs are the atomic information in DST, much like probability of single events in probability theory. By definition, $m(A)$ is the part of belief that supports exactly A but, due to lack of knowledge, does not support any strict subset of A, otherwise the mass would "move" into the subsets. In the previous example, if we had assigned mass 0.85 to proposition $\{ac \cup nc\}$ and 0.15 to $\{ac\}$ it would have meant that there is some evidence that the patient is affected by cancer but, based on current knowledge, a great part of our confidence cannot be assigned to none of the two specific propositions. Whenever we have enough information to assign all of the mass to singletons[1], DST collapses to probability theory.

Intuitively, if we want to obtain the total belief for a set A, we must add the mass of all proper subsets of A plus the mass of A itself, thus obtaining the *Belief* for the proposition A.

Definition 2. *Given the BBA in definition 1, the Belief function $Bel : 2^\Theta \to [0, 1]$ is defined as follows:*

$$Bel(A) = \sum_{B \subseteq A} m(B).$$

$Bel(A)$ summarizes all the reasons we have to believe in A based on the available knowledge. There are many relationships between $m(A)$, $Bel(A)$ and other functions derived from these; here we just highlight that $Bel(A) + Bel(\bar{A}) \leq 1 \ \forall A \subseteq \Theta$ and $1 - (Bel(A) + Bel(\bar{A}))$ is the lack of information (or the amount of doubt) about A.

2.1.2 Combination rule

If we have two BBAs defined over the same frame, which have been obtained from two independent sources of information, we can use Dempster's *combination rule* to merge them into a single one. Notice that the concept of independence between sources in DST is not rigorously defined (as it is, for example, in Bayesian theory): the intuition is that different pieces of evidence must have been determined by different (*independent*) means [5].

[1] A singleton is a set with exactly one element.

Definition 3. *Let Bel_1 and Bel_2 be belief functions over the same frame Θ with BBAs m_1 and m_2. Let us also assume that K, defined below, is smaller than 1. Then for all non-empty $X \subseteq \Theta$ the function m_{12} defined as:*

$$m_{12}(X) = \frac{1}{1-K} \cdot \sum_{\substack{A,B \subseteq \Theta: \\ A \cap B = X}} m_1(A) m_2(B) \tag{3}$$

where $K = \sum_{A,B:A\cap B=\emptyset} m_1(A)m_2(B)$, is a BBA function defined over Θ and is called the orthogonal sum of Bel_1 and Bel_2, denoted by $Bel_1 \oplus Bel_2$.

K measures the *conflict* between m_1 and m_2: the higher the K, the higher the conflict. The meaning of K can be understood from its definition, since it is obtained by accumulating the product of masses assigned to sets having empty intersection (which means incompatible propositions). Furthermore, we see that Dempster's combination rule treats conflict as a normalization factor: in practice, this means that the amount of conflicting evidence is proportionally "redistributed" to non-conflicting propositions. Later in this section, it will become evident that such a way of handling conflicting evidence can lead to counter-intuitive results in some cases.

Recall the example in (2), and suppose that we obtain evidence coming from another doctor, who is not a cancer specialist, about the variable C. Let us call m_1 the BBA in eq. (2) and m_2 the new assignment. We have:

$$m_1(X) = \begin{cases} 0.8 & \text{for } X = \{ac\} \\ 0.2 & \text{for } X = \{nc\} \end{cases} \qquad m_2(X) = \begin{cases} 0.1 & \text{for } X = \{ac\} \\ 0.9 & \text{for } X = \{ac \cup nc\} \end{cases}.$$

Since the second doctor is not a specialist, the information he provides is quite limited, and hence most of the mass is assigned to doubt. Fusing the two pieces of information according to Dempster's rule results in:

$$m_{12}(X) = \begin{cases} \frac{0.8\cdot 0.1 + 0.8\cdot 0.9}{1-(0.1\cdot 0.2)} = 0.816 & \text{for } X = \{ac\} \\ \\ \frac{0.2\cdot 0.9}{1-(0.1\cdot 0.2)} = 0.184 & \text{for } X = \{nc\} \end{cases}.$$

We see that after fusion values are not far from those already assigned by m_1: this is perfectly intuitive, since the second doctor did not bring a clear contribution to the diagnosis. Notice also that for the same reason, and for the low confidence of the first doctor about absence of cancer, little conflict is observed ($K = 0.02$).

Dempster's rule has many properties [6]; here we are mainly interested in associativity and commutativity, that is:

$$Bel_1 \oplus (Bel_2 \oplus Bel_3) = (Bel_1 \oplus Bel_2) \oplus Bel_3 \tag{4}$$

$$Bel_1 \oplus Bel_2 = Bel_2 \oplus Bel_1. \tag{5}$$

Despite its desirable properties, Dempster's rule is not idempotent; this means that observing twice the same evidence results in stronger beliefs. This is the reason why we need to introduce the hypothesis of independent sources in Dempster's combination rule. In practice, before letting a new source of information enter the system, we must always look at how the new information is collected, to ensure that we are not counting twice the same evidence. In our example, we must be sure that the doctors did not talk with each other, did not use the same technology when performing measurements, and so on.

Before moving to the next topic, it is worth pausing to discuss the way Dempster's rule manages conflicting evidence. When we combine evidence using Dempster's rule, it is assumed that masses are assigned by reliable sources, acting like oracles: the responsibility of declaring doubt is demanded to sources. When this fact is not properly accounted for, it becomes easy to reach counter-intuitive conclusions. One noticeable example of this is Zadeh's paradox [7], that can be explained by re-visiting our example: let us suppose that two doctors provide the following

BBAs, where $\{ac\}$ is the proposition "the patient is affected by cancer", $\{ap\}$ is the proposition "the patient is affected by pneumonia" and $\{af\}$ is the proposition "the patient is affected by flu":

$$m_1(X) = \begin{cases} 0.9 & \text{for } X = \{ac\} \\ 0.1 & \text{for } X = \{ap\} \\ 0 & \text{for } X = \{af\} \end{cases} \qquad m_2(X) = \begin{cases} 0 & \text{for } X = \{ac\} \\ 0.1 & \text{for } X = \{ap\} \\ 0.9 & \text{for } X = \{af\} \end{cases}$$

Not surprisingly, using Demspter's rule to merge the above assignments results in a strong conflict ($K = 0.99$), and the merged BBAs is:

$$m_{12}(X) = \begin{cases} 0 & \text{for } X = \{ac\} \\ 1 & \text{for } X = \{ap\} \\ 0 & \text{for } X = \{af\} \end{cases},$$

meaning that, based on available knowledge, the patient is affected for sure by pneumonia. Such a result is counter-intuitive at a first glance: pneumonia was assigned only a 0.1 mass by both doctors, but after fusion it becomes a certainty. Yet, if we think about doctors as oracles, as Dempster's rule does, then it becomes evident that pneumonia is the only possibility, because both flu and cancer had been excluded, respectively, by Doctor 1 and Doctor 2. Quoting A. C. Doyle, the rationale is that "when you have eliminated the impossible, whatever remains, however improbable, must be the truth" [8]. In the example, doctors were totally resolved in excluding cancer and pneumonia (assigning a zero mass equals to declare the proposition impossible), and only flu is left as a possibility by both of them.

Although many different combination rules have been proposed treating conflict in a more cautious way, Dempster's rule can be sfely used as long as the sources of information are modeled taking into account their intrinsic restrictions; in the end, if an oracle were available, information fusion would not be useful at all.

2.1.3 Belief marginalization and extension

The combination rule expressed in equation (3) is applicable if the two BBAs, m_1 and m_2, are defined over the same frame, which means that they refer to the same propositions. Whenever we need to combine BBAs defined over different frames, we have to redefine them on a common frame before the combination. This can be done by using *marginalization* and *vacuous extension*.

Definition 4. *Let m^Θ be a BBA function defined over a frame Θ, and let Ω be another frame. The vacuous extension of m^Θ to the product space $\Theta \times \Omega$, denoted with $m^{\Theta\uparrow\Theta\times\Omega}$, is defined as:*

$$m^{\Theta\uparrow\Theta\times\Omega}(X) = \begin{cases} m^\Theta(A) & \text{if } X = A \times \Omega, A \subseteq \Theta \\ 0 & \text{otherwise} \end{cases}.$$

This allows to extend the frame of a BBA without introducing extraneous assumptions (no new information is provided about propositions that are not in Θ). That said, vacuous extension is not the only possible way to extend a BBA to a larger frame: it just provides the "least informative" extension.

The inverse operation of vacuous extension is marginalization.

Definition 5. *Let m^Θ be a BBA function defined on a domain Θ; its marginalization to the frame $\Gamma \subseteq \Theta$, denoted with $m^{\Theta\downarrow\Gamma}$, is defined as*

$$m^{\Theta\downarrow\Gamma}(X) = \sum_{A\downarrow X} m^\Theta(A)$$

where the index of the summation denotes all sets $A \subseteq \Theta$ whose projection on Γ is X.

To define the projection operator, let us introduce two product frames Θ and Γ, that are obtained as the cartesian product of the frames of some variables. Formally, we have $\Theta =$

$F_1 \times F_2 \times \cdots \times F_k$ and $\Gamma = F_{S_1} \times F_{S_2} \cdots \times F_{S_z}$, where F_j is the frame of the j-th variable and S is a subset of the indices in $\{1, \ldots, k\}$. Each element of Θ will be a vector whose j-th component is a value in F_j. For instance, if $\Theta = X \times Y \times Z$ one possible element of Θ is (x_1, y_3, z_1), where $x_1 \in X$, $y_3 \in Y$ and $z_1 \in Z$. The projection operator maps each element $\theta \in \Theta$ into an element of $\gamma \in \Gamma$ by removing from θ all the components whose indices are not in S. For example, if we project the set $\Theta = X \times Y \times Z$ onto $\Gamma = X \times Z$ the element $(x_1, y_3, z_1) \in \Theta$ reduces to $(x_1, z_1) \in \Gamma$. The importance of extension and marginalization is that they allow to combine over a common frame BBAs originally referring to different frames, hence enabling us to fuse them with Dempster's rule.

2.2 The proposed multi-clue forensic framework

As we try to use the theoretical tools provided by DST to develop a framework for decision fusion, there are two main aspects that need to be discussed. The first one is how to fruitfully combine the information provided by different tools, once it is written in terms of Basic Belief Assignments. When we considered the toy problem of fusing information coming from two doctors, we assumed they would directly provide their knowledge in the form of consistent BBAs. Now that we are dealing with image forensic algorithms, the way their output is mapped to BBAs becomes of fundamental importance, and needs to be carefully discussed. Therefore, the second fundamental aspect that needs to be considered is how to convert the output provided by analysis instruments into Basic Belief Assignments. Fortunately these two topics can be treated separately, and we take advantage of this in the following: we first deal with the problem of merging the information provided by different tools, assuming it is already written in terms of BBAs. Then, starting from Section 2.3, we focus on the problem of mapping tool outputs into BBAs.

2.2.1 Modeling forensic tools and traces using DST

We now formalize the problem of merging the information that comes from different forensic tools. To begin with, we clarify the terminology: we will talk about *tools* searching for forensic *traces*. With "tool" we mean an algorithm that, given an image and a suspect region, performs a set of operations aiming at detecting the presence of a forensic trace. With "trace", we mean a property that may be present, either in the suspect region or in the rest of the image, and that can be possibly searched for in different domains (e.g., the DCT or the pixel domain). For example, when an image undergoes two JPEG codings with misaligned quantization grids, some artifacts are left both at the pixel level (inconsistent blocking artifacts) and in the DCT domain (double quantization of DCT coefficients). It becomes evident that different tools can be devised to search for the same trace in different domains, and that fusing their outputs can potentially improve the accuracy. Of course, we also have tools searching for different traces: in these cases, we will take advantage of knowing the compatibility relationships between traces (for example, the presence of one trace may imply the absence of another). We can now state the two basic assumptions behind the proposed framework for decision fusion:

- The compatibility relations among some or all the considered traces are known (for instance, we may know that two tools search for mutually-exclusive traces);
- Each tool gathers information independently of the others (i.e., a tool is never employed as a subroutine of another, and no information is exchanged between tools), and by different means (each tool relies on a different principle or effect).

These assumptions are very reasonable in the current image forensic scenario. However, as it will be shown later on (Section 2.2.4), the first assumption can be relaxed arbitrarily, at the cost of a lower performance gain with respect to the use of single tools alone. The second assumption, instead, is needed to ensure that we can fuse tools' responses using Dempster's rule. Intuitively, it means that if we observe two different tools supporting the same proposition, we are more confident than observing only one. On the other hand, if two tools searching for the same trace by exploiting the same model are available, it makes sense to discard the less reliable one, since its

contribution is limited or null. That said, and also considering that the concept of independence in DST is not equivalent to statistical independence, we believe that possible limited dependencies between algorithms would not undermine the correct behavior of our framework.

Formalization for a single tool. For sake of clarity, we start by formalizing the DST framework when only one tool is available, let us call it *ToolA*, which searches for a forensic trace called α and outputs a scalar value A. The key idea is to treat *ToolA* as a source of information about the presence of the trace α. To this aim, we define for α the frame $\Theta_\alpha = \{t\alpha,\ n\alpha\}$, where $t\alpha$ is the proposition "trace α is present" and $n\alpha$ is the proposition "trace α is not present". We model the information provided by *ToolA* about the presence of α with the following BBA over the frame Θ_α:

$$m_A^{\Theta_\alpha}(X) = \begin{cases} A_{\mathrm{T}} & \text{for } X = \quad \{(t\alpha)\} \\ A_{\mathrm{N}} & \text{for } X = \quad \{(n\alpha)\} \\ A_{\mathrm{TN}} & \text{for } X = \quad \{(t\alpha) \cup (n\alpha)\} \end{cases} \tag{6}$$

where A_{T}, A_{N} and A_{TN} are functions mapping the response of the tool A into mass assignments; as anticipated, the definition of these functions will be the subject of Section 2.3. We see that this BBA assigns a mass to every element of the power set of Θ_α; $\{(\mathrm{t}\alpha) \cup (n\alpha)\}$ is the doubt that *ToolA* has about the presence of the trace, so it refers to the proposition "trace α is either present or not". It is worth remarking the importance of allowing tools expressing lack of certainty, especially in the image forensic field were no algorithm exists that is highly reliable under any possible situation. If every tool only declares the actual degree of confidence about presence of the searched trace, this will make its contribution more valuable when it comes to be merged with others. Failing to do so, on the contrary, may result in counterproductive behaviors.

2.2.2 Introducing new tools

Suppose now that we want to introduce in our framework a new tool, *ToolB*, that satisfies the assumptions given at the beginning of this section. As we anticipated, two situations are possible: the new tool may either search for a trace that is already considered in the framework, or for a novel trace. Since Dempster's combination rule allows fusing only BBAs that are defined over the same frame of discernments, these two cases must be addressed differently.

Introduction of a tool looking for a known trace. If the trace searched for by the new tool is already present in the framework (let us call it α, consistently with Section 2.2.1), the application of the procedure in Section 2.2.1 will produce $m_B^{\Theta_\alpha}$, which can be directly fused with $m_A^{\Theta_\alpha}$ by using Dempster's rule, yielding:

$$m_{AB}^{\Theta_\alpha}(X) = \frac{1}{1-K} \cdot \begin{cases} A_{\mathrm{T}}B_{\mathrm{T}} + A_{\mathrm{T}}B_{\mathrm{TN}} + A_{\mathrm{TN}}B_{\mathrm{T}} & \text{for } X = \quad \{(t\alpha)\} \\ A_{\mathrm{N}}B_{\mathrm{N}} + A_{\mathrm{N}}B_{\mathrm{TN}} + A_{\mathrm{TN}}B_{\mathrm{N}} & \text{for } X = \quad \{(n\alpha)\} \\ A_{\mathrm{TN}}B_{\mathrm{TN}} & \text{for } X = \quad \{(t\alpha) \cup (n\alpha)\} \end{cases} \tag{7}$$

where $K = A_{\mathrm{T}}B_{\mathrm{N}} + A_{\mathrm{N}}B_{\mathrm{T}}$. This BBA contains the information about the trace α gathered by the two distinct tools. We see that conflict is non-null, and is obtained by summing the masses for propositions in which the tools provide conflicting information about the presence of the trace. It is worth repeating that before introducing a new tool into the framework, the user should understand how the tool works and ensure that it does not replicate the investigation of a tool that is already present, since this would violate the request of sources independency, and lead to na overestimation of the presence of the considered trace.

Introduction of a tool looking for a new trace. If $ToolB$ searches for a novel trace, say β, we must introduce it into the framework by defining a new frame $\Theta_\beta = \{t\beta, n\beta\}$, where the propositions have the same meaning as in (6). The response of $ToolB$ will be used to assign masses to the variable Θ_β, yielding $m_B^{\Theta_\beta}$. Since α and β are defined over different frames, $m_A^{\Theta_\alpha}$

and $m_B^{\Theta_\beta}$ cannot be fused directly. We need to introduce the common frame $\Theta_{\alpha\beta} = \Theta_\alpha \times \Theta_\beta$, so that we can (vacuously) extend both m_A and m_B to it, yielding:

$$m_{\mathrm{A}}{}^{\Theta_\alpha \uparrow \Theta_{\alpha\beta}}(X) = \begin{cases} A_{\mathrm{T}} & \text{for } X = \{(t\alpha, t\beta) \cup (t\alpha, n\beta)\} \\ A_{\mathrm{N}} & \text{for } X = \{(n\alpha, t\beta) \cup (n\alpha, n\beta)\} \\ A_{\mathrm{TN}} & \text{for } X = \{(t\alpha, t\beta) \cup (t\alpha, n\beta) \cup (n\alpha, t\beta) \cup (n\alpha, n\beta)\} \end{cases} \tag{8}$$

$$m_{\mathrm{B}}{}^{\Theta_\alpha \uparrow \Theta_{\alpha\beta}}(X) = \begin{cases} B_{\mathrm{T}} & \text{for } X = \{(t\alpha, t\beta) \cup (n\alpha, t\beta)\} \\ B_{\mathrm{N}} & \text{for } X = \{(t\alpha, n\beta) \cup (n\alpha, n\beta)\} \\ B_{\mathrm{TN}} & \text{for } X = \{(t\alpha, t\beta) \cup (n\alpha, t\beta) \cup (t\alpha, n\beta) \cup (n\alpha, n\beta)\} \end{cases} . \tag{9}$$

Equations (8) and (9) show what "vacuous extension" means in practice: for example, in the first line of (8) the mass A_{T} is assigned to the set $\{(t\alpha, t\beta) \cup (t\alpha, n\beta)\}$, which is the proposition "trace α is present, regardless of trace β". As expected, the mass assigned by *ToolA* does not bring any information about β. Application of Dempster's combination rule to these two BBAs gives us the desired combination:

$$m_{\mathrm{AB}}{}^{\Theta_{\alpha\beta}}(X) = \begin{cases} A_{\mathrm{T}}B_{\mathrm{T}} & \text{for} X = \{(t\alpha, t\beta)\} \\ A_{\mathrm{T}}B_{\mathrm{N}} & \text{for} X = \{(t\alpha, n\beta)\} \\ A_{\mathrm{T}}B_{\mathrm{TN}} & \text{for} X = \{(t\alpha, t\beta) \cup (t\alpha, n\beta)\} \\ A_{\mathrm{N}}B_{\mathrm{T}} & \text{for} X = \{(n\alpha, t\beta)\} \\ A_{\mathrm{N}}B_{\mathrm{N}} & \text{for} X = \{(n\alpha, n\beta)\} \\ A_{\mathrm{N}}B_{\mathrm{TN}} & \text{for} X = \{(n\alpha, t\beta) \cup (n\alpha, n\beta)\} \\ A_{\mathrm{TN}}B_{\mathrm{T}} & \text{for} X = \{(t\alpha, t\beta) \cup (n\alpha, t\beta)\} \\ A_{\mathrm{TN}}B_{\mathrm{N}} & \text{for} X = \{(t\alpha, n\beta) \cup (n\alpha, n\beta)\} \\ A_{\mathrm{TN}}B_{\mathrm{TN}} & \text{for} X = \{(t\alpha, t\beta) \cup (t\alpha, n\beta) \cup (n\alpha, t\beta) \cup (n\alpha, n\beta)\} \end{cases} \tag{10}$$

Notice that, at this point, we are not considering whether traces α and β are compatible or not: we will take this information into account only later on, exploiting the associativity and commutativity of Dempster's rule. Consequently there is no reason why the two tools should be conflicting, as confirmed by the fact that $K = 0$ in the above formula, since they are searching for traces that are considered "unrelated".

The procedures in Section 2.2.2 can be repeated when another tool *ToolX* becomes available. The associativity of Dempster's rule, defined in eq. (4), allows to combine directly the BBA $m_{X_{tot}}$ of the new tool with the one currently available (that takes into account all the tools already in the framework), so we will always need to extend the frame of, at most, two BBAs: this is a considerably smaller effort with respect to extending the BBA and computing the combination rule for all the tools.

Notice that using traces as basic entities instead of tools responses improves the extendability of the framework: as a matter of fact, while new tools are being released quite often, many of them search for an already known trace; if this is the case, introducing a new tool is very simple since only its BBA must be extended.

2.2.3 Managing configurations of tools

When combining multiple tools there is one practical fact that must be accounted for: it may happen that, for a given image, only part of the tools within the framework can be run, while others can't. For example, some tools may not be compatible with the image, due to its format, size, number of channels and so on. Is it possible for the analyst to handle these variants without increasing the complexity and extensiveness of the framework? DST offers a nice way to solve this issue. Given a set Θ, the following BBA, known as *vacuous* BBA, is the neutral element for Dempster's combination rule:

$$m_V^\Theta(X) = \begin{cases} 0 & \forall\, X \subset \Theta \\ 1 & \text{for } X = \Theta \end{cases}. \tag{11}$$

We see that $m_V^\Theta(X)$ is a valid BBA assigning all the mass to the whole frame of discernment, which means that no knowledge is brought about the elements of Θ. For any BBA m_X^Θ, we have:

$$m_X^\Theta \oplus m_V^\Theta = m_X^\Theta, \tag{12}$$

meaning that m_V^Θ can be fused an arbitrary number of times without modifying the available information.

Let us go back to our problem, and assume that there is a *ToolU*, searching for trace α, that cannot be run on the image under analysis. The analyst will simply write the following:

$$m_U^{\Theta_\alpha}(X) = \begin{cases} 0 & \text{for } X = \quad \{(t\alpha)\} \\ 0 & \text{for } X = \quad \{(n\alpha)\} \\ 1 & \text{for } X = \{(t\alpha) \cup (n\alpha)\} \end{cases}, \tag{13}$$

and use the framework without changes, *ToolU* will not contribute at all to the final belief about the presence of trace α. In the extreme case where none of the tools available to the analyst can handle the image, we still obtain a valid BBA, telling that no information is available about the searched traces. We point out that, despite its plainness, the above feature is not easily enabled by many machine learning techniques: every possible combination of tools would result in a different feature space, thus requiring a different training.

2.2.4 Modeling traces relationships

So far, we have considered traces as if they were unrelated to each other; as it will be shown in the following sections, this is usually not the case in real applications. This kind of information can contribute significantly to the joint interpretation of tools responses.

Suppose, for instance, that we have two traces α and β and that, due to their nature, only some of their combinations are possible. For example, it may be possible that the presence of α implies the absence of β, so, at least ideally, two tools searching for these traces should never detect tampering simultaneously. This information induces a *compatibility relation* between frames Θ_α and Θ_β, meaning that some of the elements of the cartesian product $\Theta_\alpha \times \Theta_\beta$ are impossible. Strictly speaking, these elements should have never entered the frame of discernment, because by definition such frame contains only *possible* propositions, (Section 2.1.1). However, since we may not know in advance which traces will be introduced in our framework, we need a way to include this knowledge only in the late stage of fusion, and update the frame accordingly. Fortunately, in DST we can easily model this information by using a standard belief assignment: we define a BBA on the domain $\Theta_\alpha \times \Theta_\beta$, that has only one focal set, containing the union of all propositions (i.e, combination of traces) that are considered possible, while all others have a null mass. For example the following BBA:

$$m_{comp}(X) = \begin{cases} 1 & \text{for } X = \{(t\alpha, n\beta) \cup (n\alpha, t\beta) \cup (n\alpha, n\beta)\} \\ 0 & \text{for } X = \quad \{(t\alpha, t\beta)\} \end{cases} \tag{14}$$

models the incompatibility between traces α and β. Once this BBA is specified, the simultaneous presence of α and β is no longer considered possible, and any evidence supporting it will be treated as conflicting information. Thanks to the commutative property of Dempster's combination rule, this BBA can be combined with those coming from traces in the final stage of fusion. In such a way, information about tools relationships are exploited only at the very end and hence do not hinder the extendability of the model.

Of course, we want to allow the analyst to specify traces relationships without forcing him to do that. However, the given formulation can be used also when the relationships between some of the traces are not known: it is sufficient not to put unknown propositions in the impossible set of m_{comp}, meaning that there is no clue against those propositions being possible.

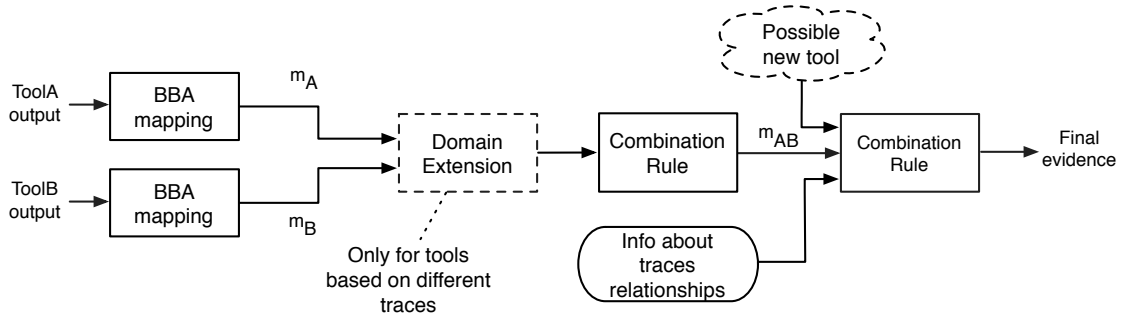


Figure 1: *Block diagram of the proposed framework. When a new tool becomes available (dashed cloud), its BBA directly enters the final stage of the fusion process, without the need to recombine information from previous tools.*

The last step of the decision fusion process consists in fusing the compatibility BBA defined above with the BBA obtained combining evidences from all the available tools, yielding a global BBA m_{FIN}. Notice that in this last application of Dempster's rule all the conflict that may arise is due to incompatibilities between traces. Although this conflict is normalized away by Dempster's rule, the value of K can be recorded and used to evaluate how "unexpected" the output of tools were. Very high values of conflict may indicate that the image under analysis does not respect the working assumptions of one or more tools. The overall decision fusion approach described so far is summarized in Figure 1 for the case of two tools.

It is worth noting that we did not need to introduce a-priori probabilities about an image being original or forged, or prior probabilities regarding the presence of traces: in a Bayesian framework, this would have been difficult to obtain.

2.2.5 Dealing with many traces: hierarchical modeling

Since the extension to novel traces is based on the cartesian product of sets, the number of variables in the framework grows exponentially with the number of different traces. However, this consideration holds only if the user is interested in a fusion approach that fully preserves the granularity of the information, meaning that, after fusing several different traces, the user wants to get the beliefs about presence/absence of each single trace separately. In practice, however, the presence of many traces is probably due to the fact that the framework is taking into account different classes of phenomena, e.g., traces related to camera artifacts, JPEG coding, geometrical inconsistencies, and so on. In such a scenario, it makes sense to treat each class of traces as a whole, and directly consider the contribution of each class when taking the final decision. This hierarchical fusion can be easily implemented within the proposed framework by using belief marginalization (see Definition 5) to collapse the contribution of several traces of the same class into a single variable, thus reducing the granularity of the information without hindering performance in terms of splicing detection. In Figure 2 we draw an example of hierarchical fusion applied to three different kinds of traces. Furthermore, compatibility among classes of traces can be introduced as well, at the end of the fusion chain.

2.2.6 Final decision rule

We are now ready to define the final output of the fusion procedure: we want to decide whether a given region of an image has been tampered with or not. To do so we consider the belief of two sets: the first one, T, is the union of all propositions in which at least one trace is detected, the second one, N, is the single proposition in which none of the traces is found (in the previous example it would be $N = (n\alpha, n\beta)$). The output of the fusion process therefore consists of two belief values, $Bel(T)$ and $Bel(N)$, calculated over the BBA m_{FIN} defined in Section 2.2.4. Optionally, we may also consider the normalization factor K (as defined in Section 2.1.2) of the

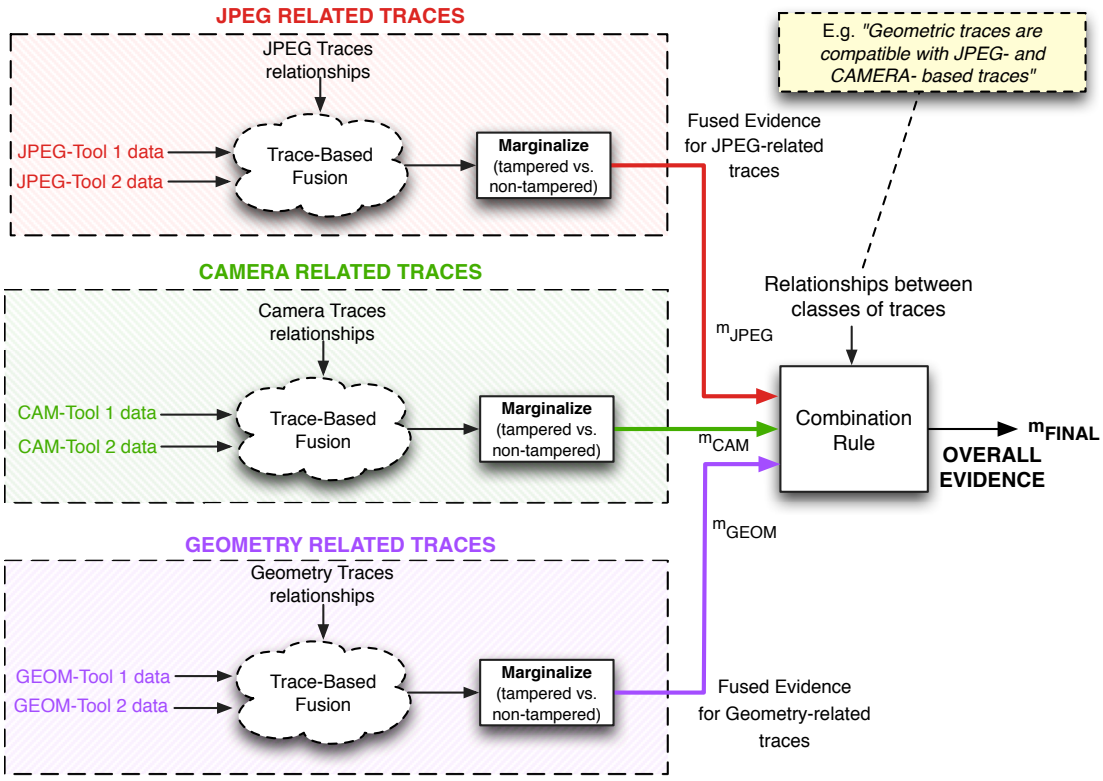


Figure 2: *Block diagram illustrating hierarchical fusion of traces of different kind. The "Trace-Based Fusion" bubble represents the schema in Figure1.*

last fusion step, involving the compatibility table. These outputs summarize the information provided by the available tools, without forcing a final decision. If a binary decision about image authenticity is required, an interpretation of these outputs must be made; the most intuitive binarization rule is to classify an image as tampered with when the belief for the presence of at least one trace is stronger than the belief for the total absence of traces, that is to say when $Bel(T) > Bel(N)$. Of course, we will probably want to meet a minimum distance requirement between the two: a Receiver Operating Characteristic (ROC) curve can thus be obtained by classifying images according to $Bel(T) > Bel(N) + \delta$, sampling δ in [-1,1].

It is worth noting that evaluating belief values is a very simple task: only elementary operations among scalar values in [0,1] must be calculated (see for example mass assignments in equation (7)), since the model is built only once for a fixed set of tools, and need to be extended only when new sources of information become available.

2.3 From tool outputs to BBAs through background information

By now we have been discussing how to manage and fuse BBAs, assuming that they are directly provided by the forensic algorithms. Needless to say, this is not usually the case: since we are working at the measurement level, each tool is expected to output a scalar value that measures the presence of the trace within the image. To be used within the framework described in the previous section, this scalar output must be "mapped" to a BBA and, as it is shown in Figure 1, this must be done for each tool separately. More formally, denoting with $\mathcal{O}_i$ the set of possible outputs of the i-th forensic tool searching for trace α, we want to derive a function

$$\mu_i : \mathcal{O}_i \to \mathcal{M}^{\Theta_\alpha}, \tag{15}$$

where $\mathcal{M}^{\Theta_\alpha}$ is the set of all possible BBAs defined on $\Theta_\alpha = \{t\alpha, n\alpha\}$.

Let us call o^i the output of the i-th tool searching for trace α; without loss of generality, we can assume that higher values of the output indicate a stronger presence of the trace. The most intuitive way to map the output to a BBA is the following:

$$m_i(X) = \begin{cases} o^i & \text{for } X = \{(t\alpha)\} \\ 1 - o^i & \text{for } X = \{(n\alpha)\} \end{cases}, \tag{16}$$

which yields a valid BBA. Of course, this approach is very rigid, because it assumes a linear relationship between the output of the tool and the belief about the presence of the trace, which is not appropriate in most cases. Let us clarify this point with an example: we ran the forensic tool[2] presented in [9] on a set of images, half of them containing the forensic trace and half of them not. Then, we collected the outputs separately for the two kinds of images and calculated their histograms, shown in Figure 3: as we can see, the output for images not containing the trace (blue bars) collapsed in the first bin, while outputs for the other class of images are more spread towards higher values. After seeing this picture, it is clearly wrong to interpret an output value of 0.5 as "uncertainty about the presence of the trace". The example above tells one trivial yet interesting thing: tool outputs must be properly *interpreted* before being merged with others. There are several reasons for such a need: tools measure different things, and their output does not necessarily have a probabilistic meaning; when they are present, probabilistic models behind tools are often approximate (e.g., they are based on over-simplified assumptions sometimes leading to anomalous behaviors); finally, even the behavior of a fixed tool may vary under different working conditions (e.g., when the analyzed region is very small or contains saturated pixel values). Therefore, passing from tool outputs to BBAs is not just a technical step, or a mere "conversion": it is a translation from a scalar measure to its interpretation in terms of belief about one proposition in the frame of discernment, which means, in our framework, presence or absence of the forensic trace. The question, then, moves to how such an interpretation should be made.

[2]This tool will be described later on in this section.

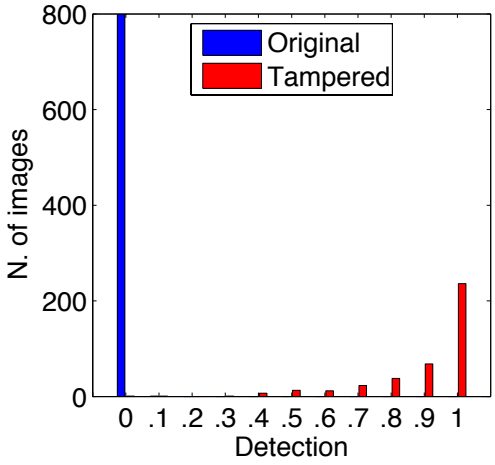


Figure 3: *Histogram of outputs obtained by running the forensic tool in [9] on a set of images, some of which containing the forensic trace searched by the tool and some not. Red bars represent the outputs obtained on images containing the trace.*

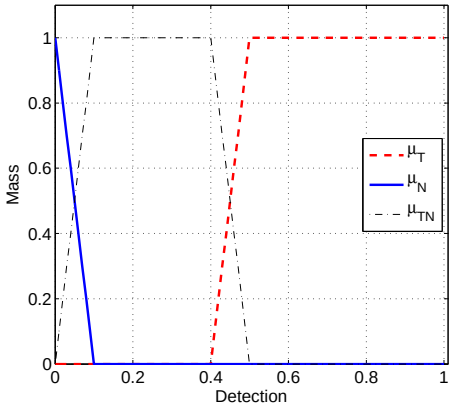


Figure 4: *A possible way of defining output mapping functions for the forensic tool in [9].*

One possible solution is to use a kind of "fuzzy-reasoning", like in [10], defining for each tool some functions allowing to map the output to masses. Formally, this can be done by writing:

$$m_i(X) = \begin{cases} \mu_{\mathrm{T}}(o^i) & \text{for } X = \{(t\alpha)\} \\ \mu_{\mathrm{N}}(o^i) & \text{for } X = \{(n\alpha)\} \\ \mu_{\mathrm{TN}}(o^i) & \text{for } X = \{(t\alpha \cup n\alpha)\} \end{cases} . \tag{17}$$

In practice, functions μ_{T}, μ_{N} and μ_{TN} together form the mapping function mentioned in equation (15). Following the example provided above, these functions could be defined as in Figure 4, where only very low values of the output are interpreted as absence of the trace, values above 0.5 are interpreted as presence of the trace, and values between 0.1 and 0.4 are characterized by a fair amount of doubt. The doubt models the fact that, based on the experiment that originated the outputs in Figure 3, we do not know how values in that range should be interpreted.

There is no doubt that this mapping is much more appropriate than the trivial one defined in (16); this is confirmed by the fact that this method has been actually employed in the works by Costanzo et al. [10] and also in some of our works [11, 12]. Yet, one may object that this approach somewhat "hides" the problem inside the definition of mapping functions, still leaving *too much work* on the user's side.

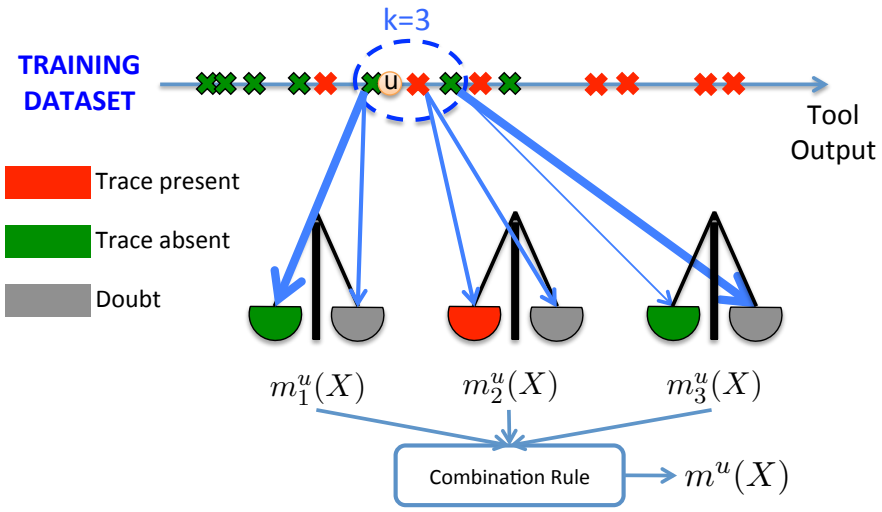


Figure 5: *Graphic representation of the proposed method for mapping tool outputs to BBAs. The weight of each arrow connecting the unseen sample u one side of the balances model the amount of mass assigned to the corresponding proposition. As we can see, the farther the training sample is from u, the more mass goes to doubt.*

2.3.1 Interpretation of tool outputs based on DST

As an answer to the above issues, we adopted a different strategy based on DST itself. The idea is still to interpret the output of a tool based on its observed behavior on a suitable number of images, but to do that in a more refined way. Let us suppose that the analyst has, for a given tool, a training set $\mathcal{T} = \{o^i : i = 1 \dots N\}$ of N training samples, where, for the i-th sample, o^i denotes the output of the tool. Each training sample belongs to one of the possible classes in $\mathcal{C} = \{C_0, C_1\}$, where C_0 is the class of images containing the searched trace, and C_1 the class of images without the trace.

As opposed to common classification problems, our goal here is not to assign an unseen sample u to one class in $\mathcal{C}$. Instead, we want to map it into a basic belief assignment over the frame Θ_α, reflecting the confidence of the tool about the presence of the searched trace. The key idea we build upon, that was first introduced in [13], is to model the elements of $\mathcal{T}$ as a source of evidence about u, and use Dempster's rule to pool the evidence. Intuitively, the closer a training sample is to u, the stronger will be the supporting evidence it provides, as shown in Figure 5.

Formally, let $\mathcal{T}_{u,k} \subset \mathcal{T}$ be the set of k training samples nearest to u according to some distance $d(\cdot,\cdot)$. Then, an element $t_i \in \mathcal{T}_{u,k}$ belonging to C_0 provides the following BBA over Θ_α:

$$m_i^u(X) = \begin{cases} \beta e^{-\gamma d(u,t_i)} & \text{for } X = \{t\alpha\} \\ 1 - \beta e^{-\gamma d(u,t_i)} & \text{for } X = \{t\alpha \cup n\alpha\} \end{cases}, \tag{18}$$

where $\beta \in (0,1)$ denotes the maximum belief we commit to a single training sample, and γ controls the width of the kernel. On the contrary, an element t_i belonging to class C_1 provides:

$$m_i^u(X) = \begin{cases} \beta e^{-\gamma d(u,t_i)} & \text{for } X = \{n\alpha\} \\ 1 - \beta e^{-\gamma d(u,t_i)} & \text{for } X = \{t\alpha \cup n\alpha\} \end{cases}. \tag{19}$$

As to the distance function, a reasonable choice is

$$d(u,t_i) = ||u - t_i||^2,$$

provided that values are well distributed within a common interval, for example [0,1].

Equations (18) and (19) deserve a comment: a sample belonging to C_0 assigns some evidence to the proposition "u comes from an image containing the searched trace" and the rest of the

evidence to the doubtful proposition "the image may or may not contain the trace". The same reasoning applies to samples belonging to C_1, as in equation (19). Notice that, when the unseen sample u is very far from t_i, this training sample will provide a BBA that is completely doubtful, instead of partitioning the mass between the two propositions $t\alpha$ and $n\alpha$. On the other hand, such a partitioning may occur after evidence pooling, when some of the k nearest neighbors belong to one class and some to the other, and they are all near to u. This situation means that the unseen sample lays in a "confused" part of the space: there are training samples near to it, but they belong to different classes.

Once the BBA assigned by each element in $\mathcal{T}_{u,k}$ has been calculated, we can use Demspter's combination rule to pool the evidence, yielding:

$$m^u(X) = \bigoplus_{i=1}^{k} m_i^u(X), \tag{20}$$

where $\oplus$ denotes the application of Dempster's orthogonal sum defined in (3) to all the m_i^u. The pooled BBA in (20) finally gives the desired interpretation of the tool output in terms of presence of the searched trace, based on training samples available to the analyst.

Compared to the simpler approaches proposed at the beginning of this section, the new method has a clear theoretical foundation that also serves as a guide for practical implementation, and requires virtually no input from the analyst (the parameters β, k and γ can be tuned with an automatic search). Yet, there is one aspect that is not directly considered: *tools reliability*. As it was discussed in Section 2.1, in DST theory it is of paramount importance to properly model the reliability of sources of information, so to avoid paradoxical situations. Looking back to equation (19) we notice that the maximum degree of certainty is mitigated by the parameter β, ensuring that we will never blindly trust one single training sample. Moreover, for tools with poor discrimination capabilities the pooled BBA (as defined in equation (20)) will likely show distributed masses among the propositions $\{n\alpha\}$ and $\{t\alpha\}$, thus accounting for the lower reliability of the tool. That said, by getting to the bottom of tool reliabilities we may be able to adapt the interpretation of outputs based on the properties of the analyzed content, and this is the object of the next section.

2.3.2 Introducing background information

Now that we have a theoretical model for interpreting tool outputs by the light of training information, we can turn the attention to choosing which information should be used for training. The goal is to understand whether there is some background information that can help interpreting the output of a tool before moving to the decision fusion stage. Let us start with a general consideration: a common feature of all forensic tools is that when a footprint becomes "less detectable", algorithms relying on that footprint become less *reliable*, meaning that they do not discriminate well between presence and absence of the trace. Therefore, if we know which are the measurable properties that affect most the performance of a detector, we could use this information to adapt the interpretation of the output, decreasing the certainty when the tool is being used under unfavorable conditions and viceversa. Giving a formal definition of the detectability of a generic footprint is beyond the scope of this work; besides, the detectability of different footprints is affected by different parameters, and a golden rule seems hard to derive. These considerations suggest that the reliability of a tool can be better investigated by using a sound experimental approach, that is, by conveniently testing the tool. To this end, we propose a procedure that the analyst may use to validate the reliability of the various tools as a function of a set of measurable parameters, so to establish if they actually impact the performance of the tools.

Identification of relevant properties. Suppose we have a set $\mathcal{F}$ of forensic tools whose goal is to tell if a given image contains a specific trace of forgery (we denote this hypothesis with $\mathcal{H}_1$) or not ($\mathcal{H}_0$). For simplicity, we assume that each tool $f \in \mathcal{F}$ outputs a score $s_f(x)$ (that

may be, for example, the probability of the presence of the tampering trace the tool is looking for), and decides for $\mathcal{H}_0$ when $s_f(x) \leq \tau$. In this way, the tool partitions the image space $\mathcal{X}$ in two regions: Λ_0, containing the images for which $\mathcal{H}_0$ is accepted, and Λ_1, defined similarly for $\mathcal{H}_1$. According to classical detection theory, the probability of correct detection and false alarm for the specific tool and a given τ are defined, respectively, as:

$$P_D^f = \int_{\Lambda_1(\tau)} p(x|\mathcal{H}_1)\ \mathrm{d}x \quad \text{and} \quad P_{FA}^f = \int_{\Lambda_1(\tau)} p(x|\mathcal{H}_0)\ \mathrm{d}x,$$

where $p(x|\mathcal{H}_0)$ is the probability conditioned to the hypothesis that the image does not contain the trace and $p(x|\mathcal{H}_1)$ denotes the opposite case.

Now, let us assume that the analyst has access to a vector of independent measurable properties $p \in \mathcal{P}$, where $\mathcal{P} = \mathcal{P}_1 \times \mathcal{P}_2 \times \cdots \times \mathcal{P}_P$. We are interested in relating the performance of each tool to subsets of $\mathcal{P}$; for simplicity, we restrict one property at a time to a subrange of its possible values $\mathcal{R} \subset \mathcal{P}_j$. To do that, we define

$$\mathcal{R}_j = \mathcal{P}_1 \times \ldots \mathcal{P}_{j-1} \times \{\mathcal{P}_j \cap \mathcal{R}\} \times \ldots \mathcal{P}_P. \tag{21}$$

In practice, $\mathcal{R}_j$ denotes the set of images whose j-th property takes value in $\mathcal{R}$. Notice that the assumption of independent properties is made so to simplify the discussion; the framework can be adapted to account for the presence of dependent properties by redefining the set $\mathcal{P}$.

Using the above notation, we can write the probability of detection and the probability of false alarm of f when the analysis is restricted to a specific set of images (those for which the parameter j belongs to $\mathcal{R}$):

$$P_D^f(\mathcal{R}_j) = \int_{\Lambda_1(\tau)\cap\mathcal{R}_j} p(x|\mathcal{H}_1)\ \mathrm{d}x, \tag{22}$$

$$P_{FA}^f(\mathcal{R}_j) = \int_{\Lambda_1(\tau)\cap\mathcal{R}_j} p(x|\mathcal{H}_0)\ \mathrm{d}x. \tag{23}$$

Equations (22) and (23) give the probabilities for a given threshold τ. By varying τ, a ROC curve is generated, that is commonly used to evaluate the discrimination capability of a detector. By taking the integral of the ROC, the Area Under Curve (AUC) is obtained and, finally, the Gini coefficient [14], denoted with ρ, can be used to summarize the performance of the tool:

$$\rho = 2 \times \mathrm{AUC} - 1. \tag{24}$$

By varying $\mathcal{R}_j$ in (21), the forensic analyst can investigate whether the performance of a tool change significantly when different subsets of $\mathcal{X}$ are considered.

Exploiting background information. Once the set of influencing properties has been determined, the problem is how to exploit them for improving the output interpretation. Interestingly, the system proposed in Section 2.3.1 can be adapted straightforwardly to account for background information. The idea is to expand the dimensionality of the "feature space", treating each influencing property as part of the problem (see Figure 6 for a graphical interpretation). Formally, we modify equation (15) as:

$$\mu_i : \mathcal{O}_i \times \mathcal{P}_1 \times \cdots \times \mathcal{P}_{N_i} \to \mathcal{M}^{\Theta_\alpha}, \tag{25}$$

where N_i denotes the number of influencing properties for the i-th tool. In other words, when the output of the tool for an image must be interpreted, the BBA is calculated not only from the output itself, but also considering the value assumed by the influencing properties on the specific image. This background information can be introduced easily by re-defining the training dataset as follows:

$$\mathcal{T} = \{t^i = (o^i, p_1^i, \ldots, p_P^i) : i = 1 \ldots N\} \tag{26}$$

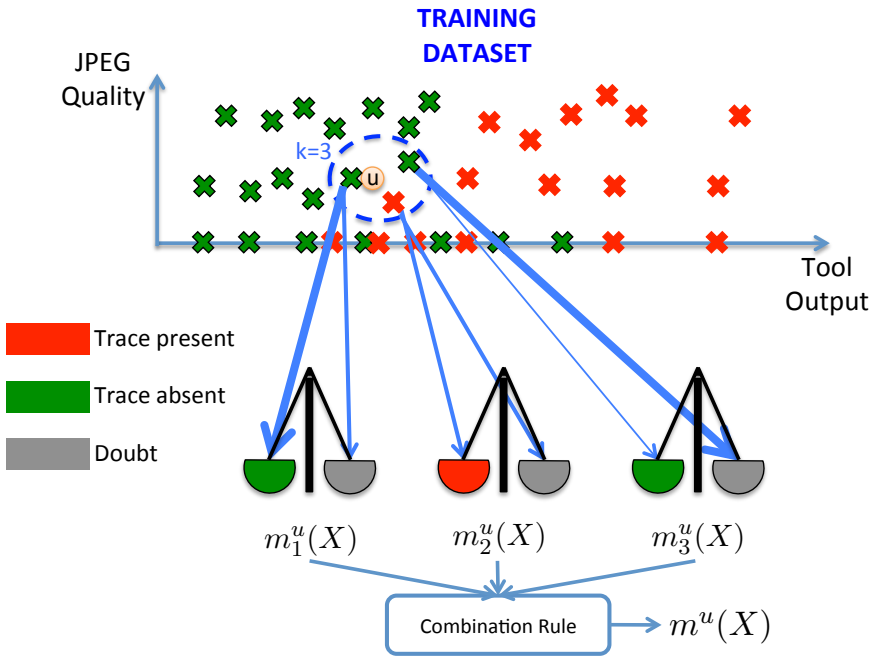


Figure 6: *Graphic representation of the proposed way to include background information in tool output interpretation. In this example, only one image property is considered (the JPEG quality factor) for clarity.*

where, for the i-th sample, o^i still denotes the output obtained from the tool and p_j^i denotes the value assumed by the j-th background property, properly scaled and normalized. Similarly, also the query vector is updated with background information: $u = (o^u, p_1^u, \ldots, p_P^u)$. No other modification is needed: given a query vector, the nearest neighbors are searched, they provide a BBA as those in equations (18) and (19), and these BBAs are merged according to equation (20) to yield $m^u(X)$.

As desired, the pooled BBA is strongly influenced by background parameters: as one or more parameters move towards unfavorable values, samples in the dataset are likely to mix between the two classes, resulting in a less informative pooled BBA. Moreover, the final BBA will be increasingly doubtful as the unseen sample moves in unpopulated parts of the space, where few training samples are available: this perfectly models the fact that the analyst does not know how much the tool can be trusted in such working conditions.

After obtaining $m^u(X)$ for each of the tools available to the analyst, the decision fusion framework can be used to fuse them together and yield a global belief about the authenticity of the image.

3 Experimental Validation and Discussion

In this section we investigate the effectiveness of the decision fusion framework presented so far. We compare it with two other possible options for decision fusion at the measurement level, using two different datasets; we also investigate the impact of the inclusion of background information in the fusion scheme. The section is structured as follows: first, we define the case study we focus on, explaining the set of forensic traces we used to create an instance of the proposed framework, together with the set of tools that can detect those traces; as a second step, we deal with the interpretation of tools outputs and their mapping to BBAs, showing which are the selected image properties and motivating their choice. Having described the framework setup, we describe the generation of the dataset together with the training and testing procedures. Finally, we compare the performance of the methods and comment them.

3.1 State of the art methods

Before going into the details of the experiments we carried out, we describe the methods we compared our framework with. The first is the simple yet widely used logical disjunction (also known as "OR rule"): the image is classified as tampered if at least one tool detects the trace it is looking for. Such a method was firstly proposed in forensics by Bayram et al. [15]. Logical disjunction is indeed one of the simplest and most widely used methods for decision fusion, and is quite well-suited to the proposed case study[3].

Several methods have been proposed for decision fusion at the feature level in image forensics [17] [18] [19] [20], but they are typically based on feature selection and are therefore not directly comparable to the method proposed in this work. On the other hand, since most methods end up using a classifier (usually an SVM), the best we can do to compare our framework with them without exiting the measurement level is to train a SVM by using the scalar output of the tools as input features, and see how the SVM performs in discriminating between tampered and original images.

Finally, limiting to the proposed framework and to the SVM-based method, we evaluate the performance obtained with and without using background information, so to investigate the benefits brought by using this kind of clue.

3.2 Reference case study and datasets

We evaluated the validity of the new DST fusion framework by focusing on the detection of splicing attacks: a portion of an image (source) is cut and pasted into another image (host), thus producing a new content that is finally saved. Because most images are stored in JPEG format, a great deal of research has been carried out for the identification and detection of traces left by splicing attacks in JPEG images, so that several tools are available to search for them. In our experiments, we fused the outputs provided by five of these tools, searching for a total of three different traces.

3.2.1 Traces and tools

To explore all the features of the proposed scheme, we chose a set of algorithms such that some of them search for the same trace, and for which some combinations of traces are not possible. Namely, we are considering the following traces (see Figure 7 for a graphical explanation):

1. *Misaligned JPEG compression* (JPNA): this trace shows up when the investigated region is cropped from a JPEG image and pasted into the target picture without preserving JPEG

[3] Actually, this approach lays somewhere in the middle between the "*abstract*" and "measurement" level, since we take the logical sum of banalized outputs, but we also properly choose how to binarize them, without blindly relying on tools mechanism. Anyway, logical disjunction is one of the most used approaches among the post-classification ones [16], so we decided to compare our method against it.

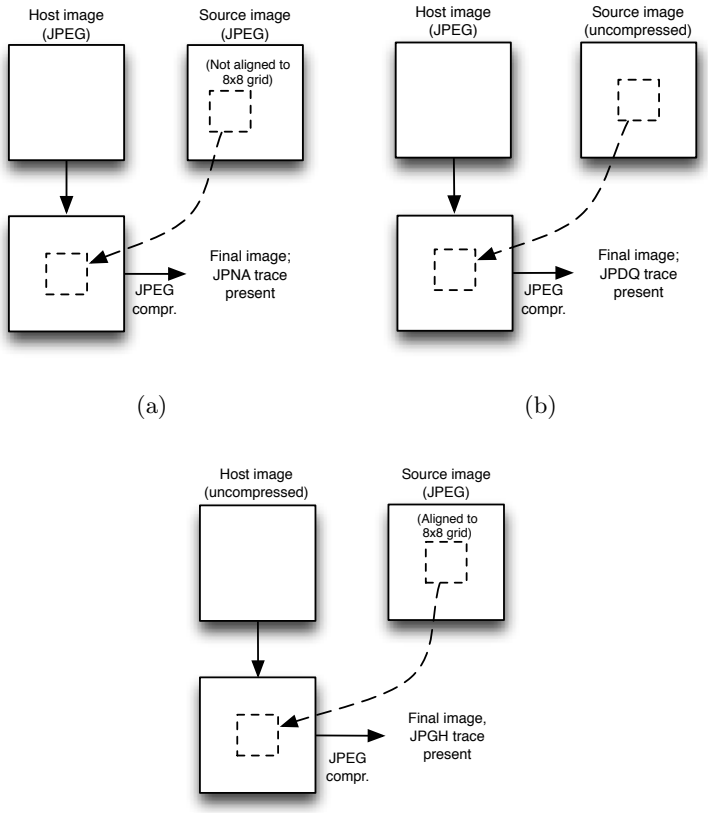


Figure 7: *In these schemes three different configurations of cut&paste attacks are reported. The attack in (a) introduces a misaligned double compression, the one in (b) introduces the double quantization effect in the untouched part of the final image and the attack in (c) introduces the ghost effect in the pasted region.*

grid alignment, performing a final JPEG compression. Therefore, pasted pixels undergo two misaligned compressions, while others do not.

2. *Double quantization* (JPDQ): when a portion of uncompressed pixels, or pixels that have been compressed according to a different grid, is pasted into a JPEG image, and the final result is saved in JPEG format, the untouched region undergoes a double compression. This causes its DCT coefficients to be doubly quantized, leaving a characteristic trace in their statistics.
3. *JPEG ghost* (JPGH): this trace appears when a region is cut-and-pasted, respecting grid alignment, from a JPEG source image into the host one (which has not been JPEG compressed). When the obtained splicing is saved in JPEG, the inserted part undergoes a second compression, while the outer part is compressed for the first time, thus introducing an inconsistency.

Given the above definitions, some combinations of traces are not possible. For example, an attack that introduces the JPDQ trace also introduces JPGH, while the contrary is not necessarily true; but, if both JPGH and JPNA are introduced, then also JPDQ must be present. These facts are best represented in tabular form, as in Tab. 1.

Trace	Possible					Excluded		
JPNA	Y	N	N	Y	N	Y	Y	N
JPDQ	N	Y	N	Y	N	Y	N	Y
JPGH	N	Y	Y	Y	N	N	Y	N

Table 1: *Detection compatibility: each column of the table forms a combination of presence (Y) and absence (N) of traces. We see that only 5 out of 8 combinations are possible.*

Now that we have introduced the traces considered in our experiments, we list the adopted forensic tools (see Tab. 2). We employed two tools looking for JPNA, namely the one by Luo et al. [21] (*ToolA*) and the one by Bianchi et al. [22] (*ToolD*); two tools looking for JPDQ, the one by Lin et al.[23] (*ToolB*) and the one by Bianchi et al. [9] (*ToolE*); and the tool by Farid that searches for ghost traces [24] (*ToolC*).

Trace	Tools
JPNA	*ToolA* [21], *ToolD* [22]
JPDQ	*ToolB* [23], *ToolE* [9]
JPGH	*ToolC* [24]

Table 2: *Coupling between traces and tools.*

As mentioned at the beginning of this section, the simple presence of a trace does not usually imply a splicing attack, but only that a common processing over the image occurred (for example, cropping a couple of rows from the top of the image would introduce a JPNA trace). Instead, *inconsistencies* in the presence of a trace through the image (i.e., high detection values for the suspect region and low for the other or vice-versa) are far more suspect. For this reason, each tool[4] is run both on the suspect region and on the remaining part of the image, and the absolute difference between the two is considered as the final output of each tool.

3.2.2 Normalization of outputs

In order to pass from outputs to BBAs, it is important to recall that formulas (18) and (19) weigh the contributions of neighboring samples in the dataset based on their distance from the observed point. Since tool outputs and reliability properties are very different in nature and can assume different ranges of values, it is important either to select a sufficiently refined distance function or to normalize them properly. We opted for the second option: in the following, we first give a brief description of how each of the selected tools works, and then define the approach we adopted to obtain a scalar output from it. We will denote by $\hat{x}_W$ the output of tool W, and by x_W its normalized version:

- *ToolA* searches for misaligned compression by measuring inconsistencies in blocking artifacts in the spatial domain. Because features are classified by using an SVM (which we trained on a separated dataset, according to the original work) we train a model supporting probability estimates [25]. The resulting outputs are well spread in the interval [0,1] and need no further processing;
- *ToolB* and *ToolE* search for double quantization traces by employing two different statistical models to analyze the histogram of the DCT coefficients of the image. Both tools provide a probability map which gives, for each 8×8 pixel block, the probability of being original (i.e.,showing double quantization) or tampered with (not showing double quantization). The final detection value is taken as the median (over the suspect region only) of

[4] *ToolC* is excluded since it already considers inconsistencies over the image.

the probability map. Being likelihood ratios, the outputs from these tools are very concentrated around 0 and 1, making their use problematic. We normalized the outputs using the following formulas: $\hat{x}_B = (\log_{10}(x_B)/15) + 1$, and $\hat{x}_E = \log_{10}(x_E)/6 + 1$ for *ToolB* and *ToolE* respectively.

- *ToolC* searches for JPEG ghost artifacts by re-compressing the image at several different qualities and taking the difference between the given image and the re-compressed one. As such, this is a tool working in the spatial domain, like *ToolA*. Ghost effect is detected when the difference is small for the suspect region and not for the rest of the image. To evaluate how much the two regions are separated, we used the KS statistic [24]. The value of this statistic can be directly used in the mapping phase without normalization;
- *ToolD* searches for misaligned double compression exploiting the fact that DCT coefficients exhibit an integer periodicity when the DCT is computed according to the grid of the primary compression. Being the shift of the grid unknown, the algorithm searches among all possible shifts the one that minimizes a specific metric (see [22] for details). We scale and invert this metric from [0,6] to [0,1] and normalize it as follows:

$$\hat{x}_D = \frac{\log_2(x_D)}{20\log_2(1.5)} + 1.$$

3.2.3 The synthetic forgery dataset

In order to generate a sufficiently large dataset, we collected a total of 630 uncompressed images representing a variety of scenes (indoor, outdoor, people, landscapes, etc.), all cropped to size 1536×1536 pixels. We considered as possible values for the size of the tampering: $64 \times 64, 128 \times 128, 256 \times 256, 512 \times 512$, and 1024×1024 pixels. Each tampering was created by pasting, in the center of the image, a region cut from another version of the *same* image. This tampering strategy creates forgeries that are virtually undetectable to the eye (see Figure 8 for some examples), and also mimics the work of an image editing expert, which would limit discontinuities along the boundary of the tampered region. By varying the way the splicing is produced, see Table 3, we generated splicings containing all the possible combinations of traces listed previously.

Figure 8: *Some sample forgeries from the synthetic dataset: the spliced region, highlighted by the dashed square, has been taken from another version of the same image, thus creating an imperceptible forgery.*

For tampered images, we let the quality of the first JPEG compression (Q_1) take values in the set $\{60, 65, \ldots, 100\}$, and the quality of the final compression is chosen as $Q_2 = \min\{Q_1 + \delta, 100\}$, where δ is chosen at random from the set $\{5, 10, 15, 20\}$.[5] Untouched images are compressed only once with $Q = \{65, 70, \ldots, 100\}$. By combining the above settings, from each uncompressed image the following files have been created:

[5]We do not investigate the case $Q2 < Q1$ because it is a setting which most image forensic cannot deal with.

Class	Procedure	Result
Class 1	Region is cut from a JPEG image and pasted, breaking the 8x8 grid, into an uncompressed one; the result is saved as JPEG.	Inner region shows JPNA trace, external region does not. *Only tool A detects this trace.*
Class 2	Region is taken from an uncompressed image and pasted into a JPEG one; the result is saved as JPEG.	Outer region shows both JPDQ and JPGH traces, inner does not. *Tools B, E and C detect this trace*
Class 3	Region is cut from a JPEG image and pasted into an uncompressed one in a position multiple of the 8x8 grid; result is saved as JPEG.	The inner region shows JPGH effect, the outer does not. *Only Tool C detects.*
Class 4	Region is cut from a JPEG image and pasted (without respecting the original 8x8 grid) into a JPEG image; the result is saved as JPEG	The inner region shows JPNA, the outer shows JPDQ and JPGH. *All tools detect this trace.*

Table 3: *Procedure for the creation of different classes of tampering in the training dataset.*

- 40 non-tampered JPEG images, by using all possible values for QF_1, and taking all possible sizes for the suspect (although not tampered) region;
- 40 forged images, by using all of the 5 possible sizes of the tampering and two random coupling for Q_1 and Q_2, thus obtaining 10 images forged according to each different procedure.

The dataset therefore consists of a total of 50400 JPEG images, half of them tampered with. Each different class of splicing consists of $25200/4 = 6300$ sample images. During the creation of the dataset, we annotated both the average value and the standard deviation of pixels in the suspect region (in the case of a color image, the image is converted to the YCbCr space and the Y channel is considered). The resulting dataset is available for downloading[6], together with the output obtained from the 5 considered tools.

3.2.4 The realistic forgery dataset

We also studied a more realistic scenario: a team of students created 70 forgeries (some examples are given in Figure 9) using common photo editing software, respecting only a constraint about JPEG quality factors (the quality factor of the final compression is always higher than the one of the host image). Students were asked to provide both tampered images (along with ground truth masks) and original ones, for a total of 136 images. Although being rather small (creating good forgeries is a time consuming procedure) this dataset is crucial to understand how well the considered frameworks generalize to unseen cases. We will refer to this dataset as the "realistic" dataset. According to a realistic scenario, this dataset is used only for testing, and training will always be performed on images of the synthetic dataset.

[6] `http://clem.dii.unisi.it/~vipp/index.php/download/imagerepository`

Figure 9: *Some sample forgeries from the realistic dataset: in the leftmost image the license plate has been pasted, while faces of celebrities have been substituted in the other two pictures.*

3.2.5 Choice of reliability properties

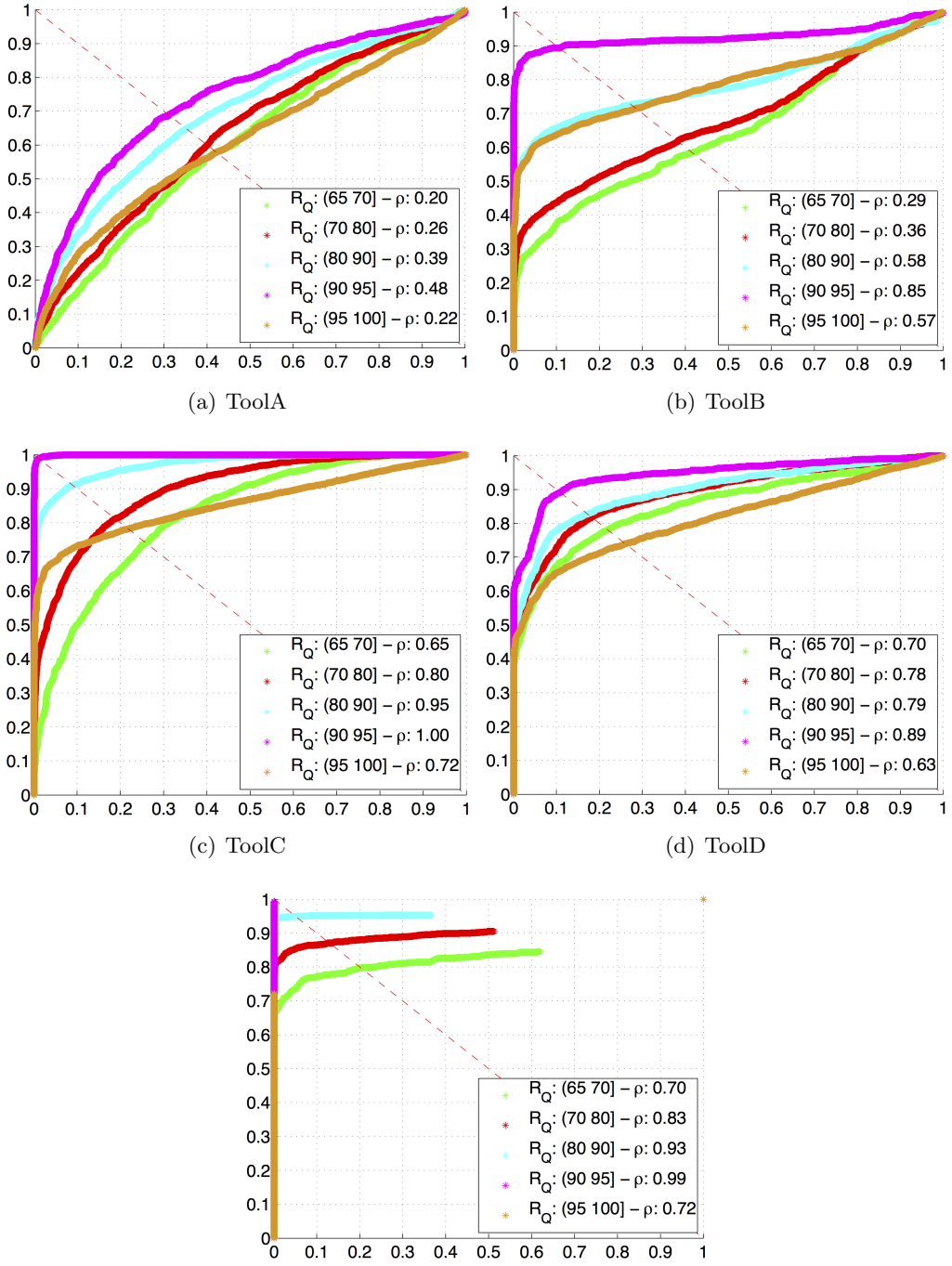
Let us now apply the BBA mapping approach proposed in Section 2.3.2 to the above case study. We define a product set of four possibly relevant properties

$$\mathcal{P} = \mathrm{Q} \times \mathrm{Z} \times \mathrm{A} \times \mathrm{S},$$

defined as follows:

- Q - *compression strength:* lossy coding after the manipulation process discards some information, thus concealing the already vanishing footprints left by the processing. Stronger compressions are against the analyst, because they erase the footprints more deeply.
- Z - *size of the analyzed region:* most forensic tools rely either on a statistical model or on the extraction and classification of some features. In both cases, working with more data results in a more reliable analysis.
- A - *average value of pixels in the analyzed region:* many forensic tools do not work well in saturated regions (i.e., having very low or very high luminance values). This holds especially for DCT-based algorithms, where the truncation errors due to saturation introduce anomalies in DCT coefficients.
- S - *standard deviation of pixels in the analyzed region:* uniform (i.e., having very low standard deviation) content yields an extremely sparse DCT representation, that can hardly lead to a reliable forensic analysis.

We used the synthetic forgeries dataset to investigate the dependency of tools performance on the above properties. Figure 10 shows the ROC curves obtained by each tool in $\mathcal{F}$ for different ranges of the property Q, along with the value of ρ calculated for each curve: we can definitely state that this property strongly influences the performance of tools in $\mathcal{F}$ and, noticeably, some tools are more sensitive than others (compare, for example, the variation of the ρ value for JPGH and JPDQ).



(a) ToolA (b) ToolB

(c) ToolC (d) ToolD

(e) ToolE

Figure 10: *ROC curves for tools in $\mathcal{F}$ for different ranges of last JPEG compression quality factor: $\mathcal{R}_Q(a, b]$ denotes the set of all images in the dataset whose last compression quality factor falls within $(a, b]$. The probability of detection P_D^f is plotted against the probability of false alarm P_{FA}^f.*

Instead of plotting similar figures for each of the investigated properties, in Table 4 we summarize the analysis for other elements of $\mathcal{P}$. We see that all the properties affect the performance of the tools and, most noticeably, not all the tools are affected in the same way. Consider, for instance, the size of the analyzed region (parameter Z): it strongly affects the performance of ToolC and ToolA but does not influence significantly ToolD. When performing a joint analysis, such an information can greatly help the analyst in reaching a correct global decision.

Tool	$\mathbf{R}_{\mathrm{Z}}^{1}$: (0,64]	$\mathbf{R}_{\mathrm{Z}}^{2}$: (64,128]	$\mathbf{R}_{\mathrm{Z}}^{3}$: (128,256]	$\mathbf{R}_{\mathrm{Z}}^{4}$: (256,512]	$\mathbf{R}_{\mathrm{Z}}^{5}$: (512,1024]
ToolA	0	0.08	0.21	0.31	0.40
ToolB	040	0.39	0.36	0.31	0.21
ToolC	0.63	0.67	0.71	0.75	0.80
ToolD	0.74	0.75	0.74	0.73	0.72
ToolE	0.37	0.62	0.72	0.75	0.78

	$\mathbf{R}_{\mathrm{A}}^{1}$: (0,30]	$\mathbf{R}_{\mathrm{A}}^{2}$: (30,60]	$\mathbf{R}_{\mathrm{A}}^{3}$: (60,150]	$\mathbf{R}_{\mathrm{A}}^{4}$: (150,230]	$\mathbf{R}_{\mathrm{A}}^{5}$: (230,255]
ToolA	0.15	0.19	0.23	0.14	-0.23
ToolB	0.09	0.35	0.38	0.25	0.19
ToolC	0.49	0.68	0.73	0.62	0.20
ToolD	0.58	0.78	0.80	0.60	0.36
ToolE	0.50	0.63	0.70	0.54	0.04

	$\mathbf{R}_{\mathrm{S}}^{1}$: (0,5]	$\mathbf{R}_{\mathrm{S}}^{2}$: (5,10]	$\mathbf{R}_{\mathrm{S}}^{3}$: (10,15]	$\mathbf{R}_{\mathrm{S}}^{4}$: (20,40]	$\mathbf{R}_{\mathrm{S}}^{5}$: (40,60]
ToolA	0.07	0.13	0.18	0.21	0.30
ToolB	0.28	0.28	0.34	0.38	0.33
ToolC	0.51	0.69	0.70	0.73	0.74
ToolD	0.46	0.65	0.76	0.79	0.80
ToolE	0.31	0.60	0.65	0.71	0.73

Table 4: *Impact of parameters Z, A and S on the performance of five image forensic tools. Intervals are chosen so to emphasize extreme values for each parameter.*

Since reliability parameters are very different in nature, it is necessary to normalize their values before using them. We used the following order-preserving functions to normalize them in the interval [0,1] (W denotes the normalized version of $\hat{\mathrm{W}}$):

- Size of the suspect region: denote with X and Y the height and width of the image, then:

$$\mathrm{S} = \frac{\log_2(\sqrt{X * Y}) - 3}{6}.$$

- Compression Quality Factor: $\mathrm{QF} = \hat{\mathrm{QF}}/100$.
- Average pixel value: $\mathrm{AVG} = \hat{\mathrm{AVG}}/255$.
- Standard deviation (STD): for natural images, the standard deviation will unlikely assume values higher than 100. Therefore, the scaled parameter is obtained as: $\mathrm{STD} = \hat{\mathrm{STD}}/100$.

3.3 Training procedure

For all the fusion techniques used in the tests we need to run a training phase. For creating train and test datasets, we divided the synthetic forgery dataset in two parts, with 80% of the images

used for training and 20% for testing. The whole procedure is repeated 10 times to increase the statistical significance of the experiment. It is worth noting that, in the proposed framework, training affects only the BBA mapping phase, so it is performed separately for each tool. On the contrary, an SVM cannot be trained separately for each tool: it must "see", for each training image, the joint outputs coming from all tools, so to learn how to fuse them. Accordingly, training the SVM by providing it forged images containing all possible combination of traces would not be realistic, since it would require a dataset whose size grows exponentially with the number of traces. We find it more reasonable to limit the training dataset to original images and images containing all the forensic traces, i.e., images belonging to "Class 4" (see table 3). Of course, this restriction is applied to all the tested techniques. We also point out that tool outputs and values of reliability properties have been normalized in the same way (Section 3.2.1) before being used with all the methods. In the following, more specific information about the training procedures for each method are given.

- *SVM fusion.* We used a radial basis function (RBF) kernel, whose parameters C and γ are selected through a grid search. The search was repeated independently two times, one for the SVM that is trained with both tool outputs and background information ($C = 4, \gamma = 4$), and one for the SVM trained only with tool outputs ($C = 256, \gamma = 0.5$).
- *DST fusion.* The output interpretation procedure presented in Section 2.3.2 was used for mapping tool outputs to BBAs. As for the SVM, the experiment was repeated twice, once including background information and once not. As to the parameters for the BBA mapping, we ran a grid search and chose, for both the experiments, $\beta = 0.8$, $\gamma = 8$ and $k = 12$.
- *OR-based fusion.* Since ROC curves are used to compare the various methods, we need to train an *aggregate* ROC for the five algorithms, which represents their behavior in terms of probability of detection (p_D) and false alarm (p_{FA}) after being combined with the OR operator. To obtain these curves, we uniformly sampled (with precision 10^{-3}) the ROC of each algorithm, considering only images that satisfy the corresponding working assumptions, as reported in Table 3. For each algorithm we saved the threshold associated with each p_{FA}. During the test phase, given a target overall probability of false alarm $\hat{p}_{FA}$, we chose for each algorithm the threshold corresponding to a probability of false alarm of $\hat{p}_{FA}/5$, and we used that threshold to binarize the output. Binarized outputs for each image were then combined with the OR operator, giving the final classification, that allows drawing a point of the overall ROC.

Concerning the realistic dataset, as we said, this dataset is used only for testing, while training is performed using synthetic images. When experiments are carried out on the realistic dataset, 100% of the synthetic dataset is used to generate the training set, still according to the rules described above.

3.4 Results

The five forensic tools were run on the datasets, gathering the selected reliability properties from the images, and their outputs were combined by the different fusion methods. We use ROC curves to compare the performance, also calculating the Gini coefficient to allow a more compact evaluation. Each ROC curve is obtained by averaging the results obtained on the 10 train-test selections; we also plot uncertainty bars showing the maximum and minimum probability of detection retained for different false alarm probabilities. For sake of clarity, we separately comment the results obtained for the synthetic dataset and those obtained for the realistic dataset.

Results on the synthetic dataset. First of all, we show in Figure 11 the ROC curve obtained by executing each stand-alone forensic algorithm on the whole dataset. The reader will probably be surprised by the poor performance obtained by single algorithms, but they are perfectly reasonable since each algorithm is used to analyze all classes of images, not only those that are detectable with it. This approach is close to reality: a real analyst does not know in advance which kind of tampering could have been performed on the image under analysis. On

the contrary, when a forensic algorithm is developed and evaluated in scientific literature, it is typically tested with images that are either original or tampered with in a "detectable way". Although being useful to evaluate the discriminative power of a specific footprint, this approach may lead to a rather optimistic evaluation of tool performance.

As to the comparison between different decision fusion frameworks, Figure 12 shows the results obtained with the three methods described earlier. We can state that the DST method providfes slightly better performance compared to the SVM: this is an encouraging result, especially if we consider that both training and test forgeries are synthetically generated in this dataset, and that we have a high ratio of training examples versus features (about 10,000 training examples in front of 9-dimensional, normalized features). Interestingly, logical disjunction also shows good performance on this dataset. The most evident conclusion we can draw from this experiment is that all fusion methods guarantee a sensible performance gain compared to single tools, thus confirming that multi-clue analysis helps moving towards a more comprehensive forensic analysis.

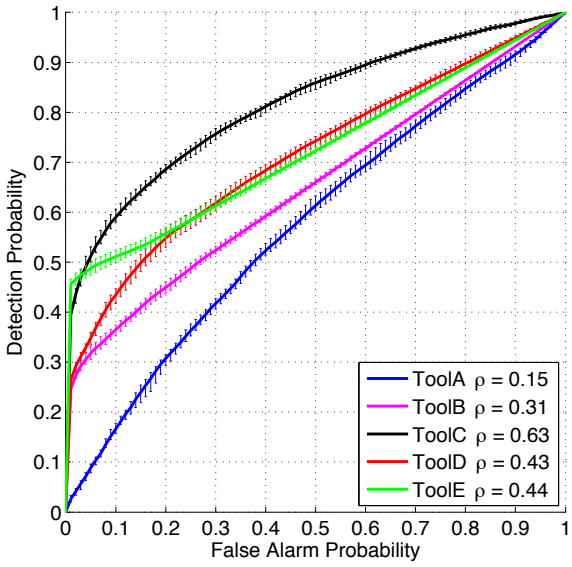


Figure 11: *Performance obtained by each forensic tool alone on the synthetic test dataset. Although tools do not require a training phase, uncertainty bars are reported because their evaluation is repeated on different selections of the test dataset.*

Let us now focus on the contribution brought by background information to the DST- and SVM- based methods. Figure 13 shows the performance of the two methods with and without the use of such an information. We can see that by including background information the analyst yields a clear advantage, regardless of the chosen framework. This gain gets even more interesting if we consider that including background information has a negligible cost in terms of complexity, at least up to a small number of properties. On the other hand, the analyst should beware of selecting a vast set of influencing parameters, since this can potentially expose the framework to the "curse of dimensionality".

Results on the realistic dataset. Focusing on the realistic dataset, Figure 14 shows the performance of single tools. We can see a variation in performance compared to the curves in Figure 11, a fact that is not surprising because of the different nature of hand-made forgeries: small size and irregular shape of the tampered area, post-processing following the cut-&-paste operation, and possibly other factors affect the forensic analysis.

The most interesting results are those obtained by the decision fusion methods on the realistic dataset. As Figire 15 shows, the DST method strongly outperforms both the OR- and SVM-based approaches on such a dataset. The most evident fact is that the SVM-based method seems

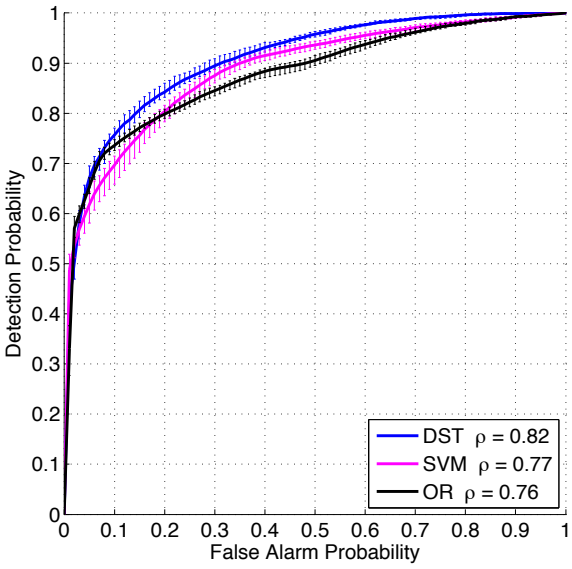


Figure 12: *Performance obtained on the synthetic dataset by the proposed fusion framework and by the other methods.*

to suffer significantly from the deep mismatch between the characteristics of the training and testing datasets. In fact, we believe that such a mismatch is unavoidable in practical situations, because it would be very hard to create a huge hand-made realistic dataset, resembling all possible kinds of operations the forger could do. It is much more reasonable, in our opinion, to define formally and unambiguously an automatic method to generate the training set, and then use the trained fusion system on realistic data. This is actually what is done with the DST-based framework.

We still have to evaluate the impact of background information on the performance obtained on the realistic dataset: as shown in Figure
reffig:resbnbReal, also from this point of view, the DST-based method is preferable, since it successfully exploits the presence of background information. On the other hand, the SVM method seems to be penalized by background information for low false-alarm rates, that are by far the most important in a forensic scenario. It is probably useful to remark that there are no differences in the "feature vectors" provided to the DST and SVM methods, meaning that both tool outputs and values of reliability properties are normalized in the same way, as explained in Section 3.2.5. Therefore, we should rather refer again to the mismatch between training and testing examples to explain the difference in the impact of background information on the two frameworks.

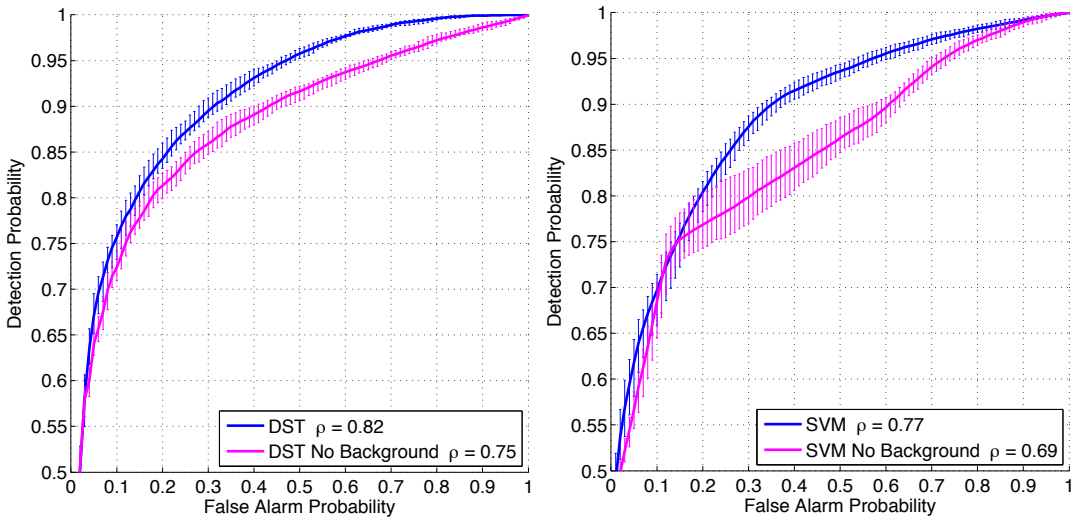


Figure 13: *Comparison between performance of the background information aware fusion methods and their simpler version, that does not use such an information; results refer to the synthetic dataset.*

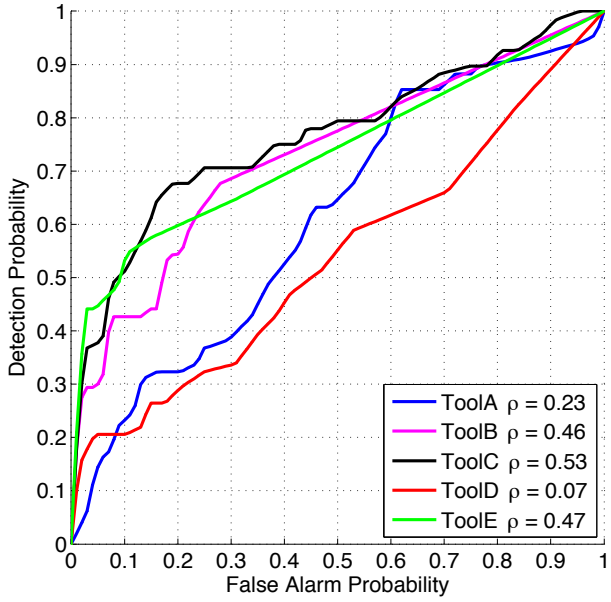


Figure 14: *Performance obtained by each forensic tool alone on the realistic test dataset. Differently from Figure 11, uncertainty bars are not present because the whole realistic dataset is always used as the test set.*

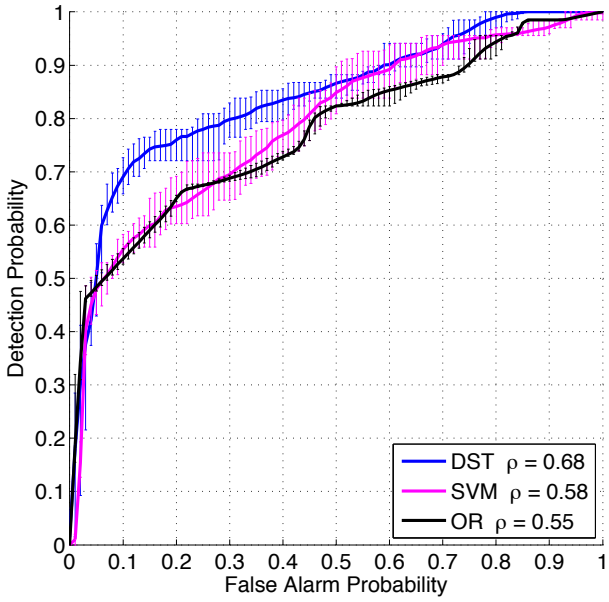


Figure 15: *Performance obtained on the realistic dataset by the proposed fusion framework and by the other considered methods.*

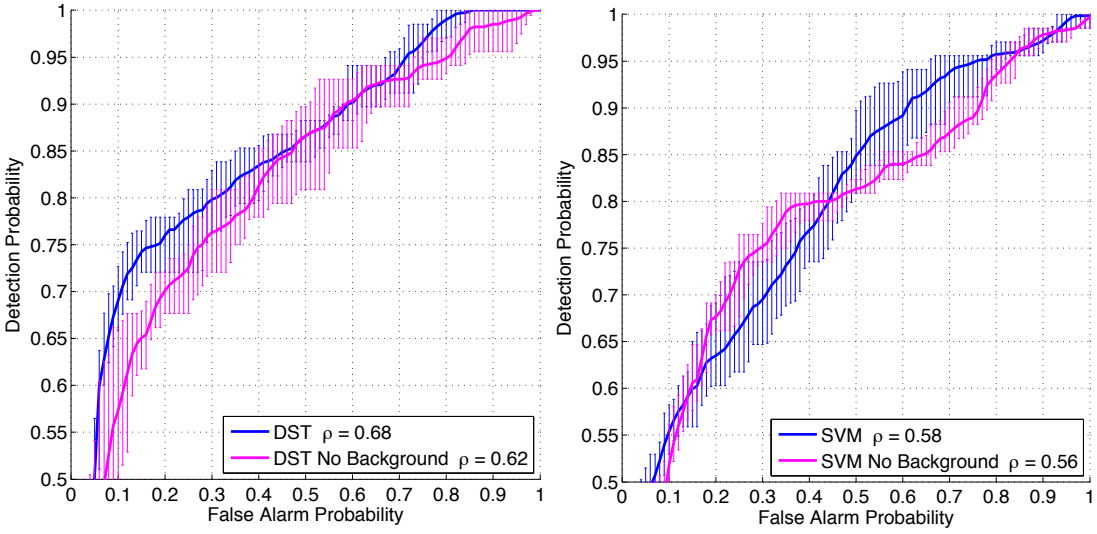


Figure 16: *Comparison between performance of the background information aware fusion methods and their simpler version, that does not use such an information; results refer to the realistic dataset.*

Figure 17: *Image related to case study 1.*

3.4.1 Some noticeable case studies

Besides presenting the results obtained on the synthetic and realistic datasets as a whole, it is interesting to isolate some noticeable case studies where the DST and SVM methods provide significantly different results. In order to select such examples, we focused on those images of the realistic dataset for which, using the threshold that give for both schemes a false alarm probability of 10%, the DST method provides a correct classification while the SVM does not. As the significant distance between the two ROC curves in Figure 15 suggests, at this "working point" there are several cases for which the DST method outperforms the SVM, specifically 16 out of 136 total images. Among these, we select and comment 5 of them in the following. For each case study, we report:

- the image and the suspect region on which algorithms have been run;
- the properties associated to the analyzed region that have been fed to the SVM and BBA-mapping modules;
- the output of forensic tools, and their interpreted version (available only for the DST method);
- the final, scalar output of the DST and SVM fusion frameworks.

Case study 1 (true negative) In the first case study (Figure 17), the DST method correctly classifies the image as original (the scalar output is 0.243), while the SVM labels it as tampered with (scalar output: 0.652). Table 5 reports the values of relevant properties and the output of each forensic tool. We denote with $m(\mathrm{T})$, $m(\mathrm{N})$ and $m(\mathrm{D})$ the interpretation provided by the BBA-mapping module for propositions "the trace is present", "the trace is absent" and doubt respectively.

The most interesting fact of this case study is the strong conflict between the two tools searching for the JPNA trace (namely, Tool A and Tool D), highlighted in red in the table. This is already evident looking at the output (first line of the table), but it becomes even more dramatic if we consider the interpretation resulting from the BBA-mapping module. The direct consequence of such a conflicting situation is that belief stemming from these two tools cancels, and is re-distributed based on the information available from other tools. Since none of the three remaining tools found any trace, the image is thus classified as original.

Compression	Size	Avg. value	St. Dev.
90	1413	74.45	38.50

	Tool A	Tool B	Tool C	Tool D	Tool E
Output	0.53	0.00	0.17	0.00	0.03
m(T)	1.00	0.01	0.00	0.00	0.12
m(N)	0.00	0.99	1.00	1.00	0.88
m(D)	0.00	0.00	0.00	0.00	0.00

Table 5: *Background properties values, tool outputs and their mapping to BBAs (case study 1).*

Compression	Size	Avg. value	St. Dev.
85	100	129.09	22.66

	Tool A	Tool B	Tool C	Tool D	Tool E
Output	0.23	0.00	0.43	0.03	0.08
m(T)	0.01	0.00	0.01	0.00	0.48
m(N)	0.99	1.00	0.99	1.00	0.52
m(D)	0.00	0.00	0.00	0.00	0.00

Table 6: *Background properties values, tool outputs and their mapping to BBAs (case study 2).*

Case study 2 (true negative) Also in this case (Figure 18 and Table 6), the DST method correctly classifies the image as authentic (output value is 0.005), while the SVM detects tampering (0.713). The most relevant aspect of this case study resides in the BBA-mapping: notice that while the output of Tool C is 0.43, a value that is near to the decision threshold for that tool [24], the interpretation of such a value is definitely towards absence of the trace (values are highlighted in red). This is likely due to the fact that the resolution of the image is rather low, so that the suspect region consists of just a few hundreds pixels. For such small regions, it is not surprising to reach higher values of the KS statistic employed by Tool C also for untouched regions.

Case study 3 (true positive) We now consider a case (Figure 19 and Table 7) where the DST framework correctly detects tampering (output is 1.000) while the SVM wrongly labels the image as authentic (0.086). Once again, conflicting information plays a fundamental role: we see that tools searching for trace JPDQ (ToolB and ToolE) provide totally conflicting outputs. The conflicting belief is thus redistributed across plausible assignments and, since the Tool C detected the JPGH trace, and the presence of only that trace is sufficient to declare the region tampered (according to Table 1), the final belief supports totally the tampering hypothesis.

Case study 4 (true positive) The last case study (Figure 20 and Table 8) highlights an important feature of the DST framework. As we can see from Table 8, in this case only Tool C detects traces of tampering, while all the other tools are highly confident that the trace they are looking for is not present. As we explained in previous sections, this fact should not lower the belief of the analyst about the presence of tampering: it may well be the case that only one trace was left during forgery creation. Therefore, as long as the combination with absence/presence of other traces is plausible, detecting that trace is sufficient to label the image as tampered with. This is exactly the case at hand, since the solely presence of trace JPGH is plausible (according to Table 1). The DST output for this image assigns 1.00 to the proposition "the image is tampered".

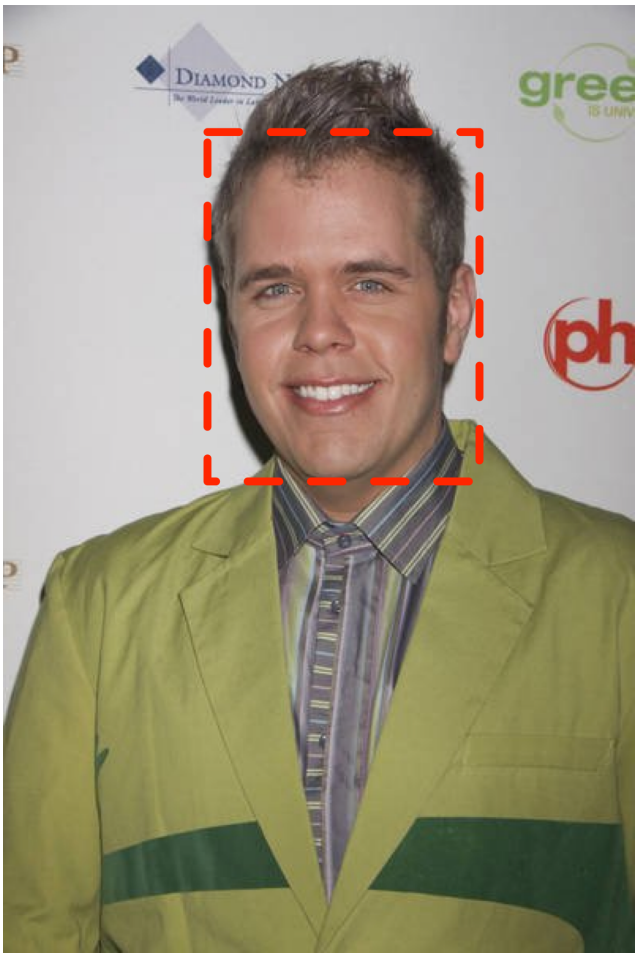


Figure 18: *Image related to case study 2.*

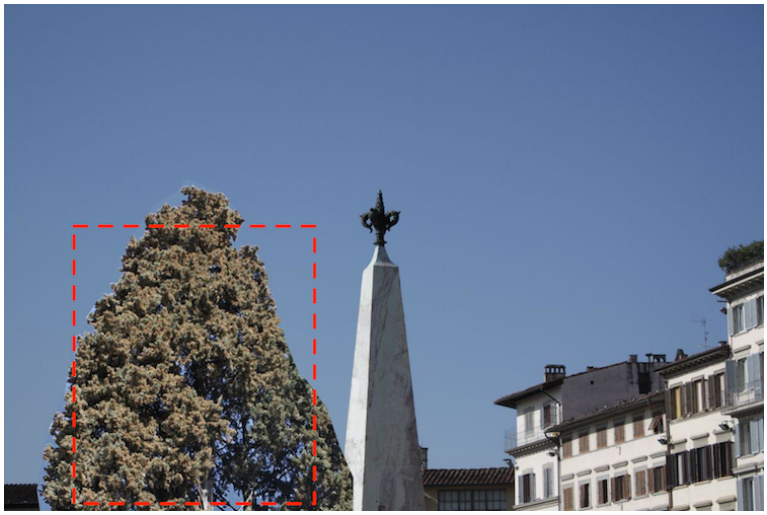


Figure 19: *Image related to case study 3.*

Compression	Size	Avg. value	St. Dev.
75	377	89.95	39.94

	Tool A	Tool B	Tool C	Tool D	Tool E
Output	0.01	0.99	0.65	0.07	0.00
m(T)	0.00	1.00	1.00	0.00	0.00
m(N)	1.00	0.00	0.00	1.00	1.00
m(D)	0.00	0.00	0.00	0.00	0.00

Table 7: *Background properties values, tool outputs and their mapping to BBAs (case study 3).*

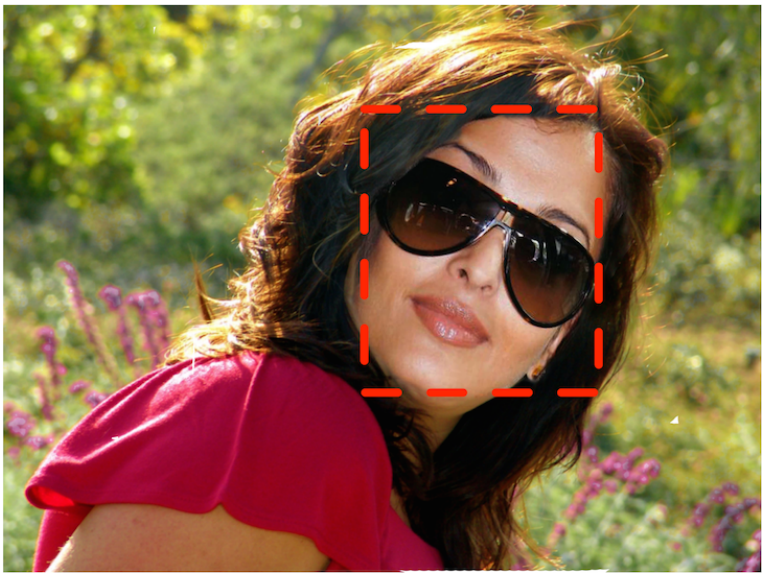


Figure 20: *Image related to case study 4.*

On the other hand, the SVM probably let other tools output "smoothen" its decision, ending up with a final score of 0.436.

3.4.2 Comments

Based on the experiments described above, we can state that the proposed method is preferable in realistic conditions because it is more "robust", meaning that differences between the training and testing datasets have a smaller impact on performance. It should also be noted that, in contrast to SVM, the training phase of the DST method treats each tool separately. This fact has several advantages, most noticeably the possibility of adding new tools without re-training the whole system, and the possibility to select different sets of influencing parameters for each tool (although this was not necessary in the case we considered here). On the other hand, the weakest point of the proposed framework resides in the specification of compatibility relationships, since the number of combinations grows exponentially with the number of different traces, as discussed in Section 2.2.5. However, as we pointed out, this fact holds only when the analyst wants to maintain a full granularity of the information. This means, besides evaluating the final belief as we did in the previous experiments, to be able to compute the belief about the presence of each single trace separately.

Compression	Size	Avg. value	St. Dev.
100	1100	97.47	57.08

	Tool A	Tool B	Tool C	Tool D	Tool E
Output	0.01	0.00	0.80	0.08	0.00
m(T)	0.00	0.02	1.00	0.00	0.00
m(N)	1.00	0.98	0.00	1.00	1.00
m(D)	0.00	0.00	0.00	0.00	0.00

Table 8: *Background properties values, tool outputs and their mapping to BBAs (case study 4).*

4 Unsupervised, multi-clue, forgery localization

One important fact that remained slightly hidden in the previous sections is that the framework described so far requires that the analyst selects the suspect region within the image. This step is crucial, since each tool is expected to output a single scalar value, which is obtained by comparing in some way the selected region with the rest of the image. In some cases, this kind of intervention is not practical, for several reasons: i) the user can hardly suspect about a region where something was hidden; ii) when a huge amount of images have to be analyzed, accurate inspection can be expensive; iii) the results produced by tools may vary even significantly when the same object is selected in different ways, and no golden rule exists in principle: an "abundant" selection may contain pixels from the background, while a "conservative" selection may result in small regions, leading to a less reliable statistical analysis. Based on these considerations, it would be desirable to develop *unsupervised* tools that allow forgery *localization*, e.g. by producing a probability map that associates to each (block of) pixel the probability of being tampered.

4.1 Prior art

Unsupervised forgery localization has been tackled with in different ways in recent literature. A first class of methods looks for the presence of tampered objects by decomposing the image under analysis into subparts. In region-wise approaches, the image is first segmented into homogeneous regions and then each region is analyzed separately [26]; in block-wise approaches, the image is split into sliding square windows, and each block is processed independently. Inconsistencies in the presence or the absence of specific footprints related to acquisition, coding, or editing within one or more sub-parts of the image indirectly reveal that some processing has been applied on a particular region of the image [27, 28]. Concerning the limits of these methods, in the region-wise approach very often the segmentation does not produce reliable results without a priori information about the possible tampered area. In the block-wise approach, usually a sufficiently large portion of the image (e.g. a $B \times B$ block, with $B \geq 100$) is needed for a reliable statistical analysis of the footprint, so that only a coarse grained localization of tampering is possible.

A second class of unsupervised tamper localization algorithms is represented by forensic schemes designed to localize in an automatic way the tampered regions with a fine-grained scale of $B \times B$ image blocks (where usually $B = 8$), assuming to have no information on the position of possibly manipulated pixels. The output of these methods is a likelihood map indicating for each pixel (or small block) its probability of being tampered.

An important limit of the previous approaches is that they are based on the observation of a single forensic trace. In practical scenarios, the simultaneous analysis of different footprints could improve tampering detection and localization.

In the rest of this section we describe a way to extend the DST multi-clue detection framework to tampering localization. As we will see, doing so will requite some modifications of the general approach and the inclusion within the fusion framework, of geometric semantic-based information. In fact, this is a crucial step to improve the accuracy of the forgery localization map.

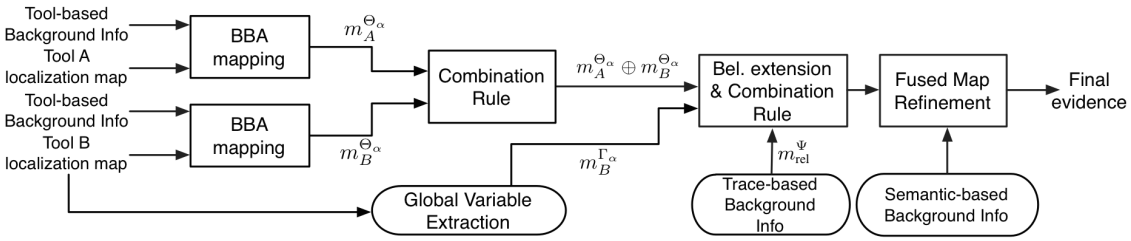


Figure 21: *Block diagram with the DST framework for forgery localization.*

4.2 The proposed method

The most intuitive approach to extend the framework described in Section 2 to forgery localization would be to apply the forensic tools to image blocks (also called "analysis block", from now on) so to form a map gig ing the output of the forensic tool for each block. The we could simply apply the DST fusion procedure separately to each element of the map. In spite of simplicity, this procedure is potentially misleading because of the nature of forgery localization tools. Indeed, the accuracy of forgery localization tools is strongly affected by the local properties of the image: for example, very smooth or saturated regions are critical for many tools (see, for example, [9, 29]), so that values assumed by the map in those regions are less reliable. As a consequence, attention must be paid to properly interpret the local output of the tools.

An overall outline of the framework that we developed to avoid such a problem is depicted in Figure 21.

4.2.1 BBA mapping.

Given a forensic trace α, and similarly to what we did in Section 2, we define the set $\Theta_\alpha = \{t\alpha,\ n\alpha\}$, where $t\alpha$ is the proposition "trace α is present in the analysis block" and $n\alpha$ is the proposition "trace α is not present in the analysis block". We model this *local* information provided by the tool τ with the following BBA over the frame Θ_α:

$$m_\tau^{\Theta_\alpha}(X) = \begin{cases} \mathrm{L}_\tau(i) & \text{for } X = \{(t\alpha)\} \\ \mathrm{N}_\tau(i) & \text{for } X = \{(n\alpha)\} \\ \mathrm{D}_\tau(i) & \text{for } X = \{(t\alpha) \cup (n\alpha)\} \end{cases} . \tag{27}$$

In the above equation $\mathrm{L}_\tau(i)$, $\mathrm{N}_\tau(i)$ and $\mathrm{D}_\tau(i)$ are scalar values obtained by interpreting the output of the tool in the i-th analysis block. It is here that *tool-based* background information enters the picture: besides considering the value of the localization map in the position of block i, some local properties of the image are evaluated (e.g., mean value or variance of pixels in the analysis block i) and used to determine $\mathrm{L}_\tau(i)$, $\mathrm{N}_\tau(i)$ and $\mathrm{D}_\tau(i)$, following the same procedure we described in Section 2.3. This stage of the fusion process is represented in the left-most side of Figure 21 ("BBA mapping" blocks).

Global variables. There is another fundamental difference between forgery detection and forgery localization tools. Independently from the analysis domain (e.g., pixel or DCT domain), unsupervised forgery localization tools typically assume that the signal under analysis is the mixture of two components: one component deriving from parts of the image that were manipulated, and one deriving from unaltered parts [23, 9, 22]. A statistical model is defined for each component, and the parameters of the models are estimated from available data. Finally, each (block of) pixel(s) is assigned a probability of belonging to each model, thus producing a forgery localization map, like the one in the center of Figure 22. However, when for some reason the two components are not correctly separated, the produced localization map is practically useless, although it assigns a sensible value to each region (right hand of Figure 22). A simple yet effective way to understand whether the tool managed or not to separate the two components is to analyze the produced localization map as a whole: when the components are not separated, the whole

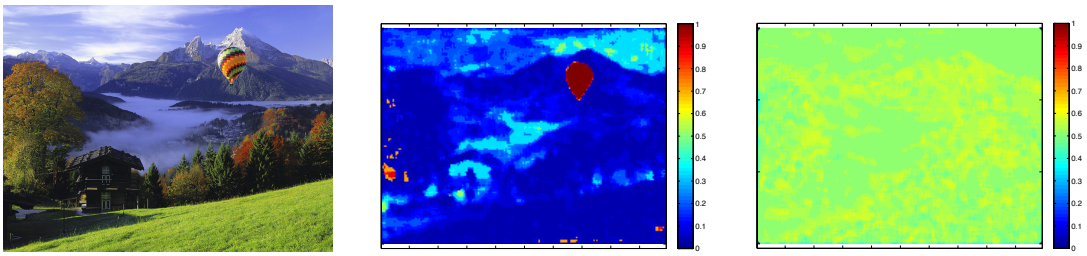


Figure 22: *A forged image (the baloon is pasted) and the forgery localization map obtained with the tool in [9] on the tampered file (center plot) and on a re-compressed version of the tampered file (right-most plot). As we can see, in the latter case the map is not discriminative as it takes values near to 0.5 everywhere; on the contrary, the same value in the center plot clearly characterizes not-tampered regions.*

map takes values in a narrow range, meaning that all pixels belong to the same component, while the opposite happens when two components are separated (compare the two maps of Figure 22 for an explicative example).

The above discussion suggests that we cannot simply interpret elements of the localization map as "stand alone small blocks", we must also model the global information that is obtained from the localization map as a whole. In order to do that, we also introduce for each forensic trace a *global* variable. Taking again the general forensic trace α as reference, we define the frame $\Gamma_\alpha = \{T\alpha, N\alpha\}$ where $T\alpha$ is the proposition "the two components related to α were separated" while $N\alpha$ has the opposite meaning. After running a localization tool searching for α, a BBA over Γ_α must be defined. We are not forced to give a binary interpretation: indeed the border between the two cases is not always sharp. Hence, for a generic tool τ, we model this information through the following BBA:

$$m_\tau^{\Gamma_\alpha}(X) = \begin{cases} (1-\mathrm{W}_\tau)\mathrm{G}_\tau & \text{for } X = \{(T\alpha)\} \\ (1-\mathrm{W}_\tau)(1-\mathrm{G}_\tau) & \text{for } X = \{(N\alpha)\} \\ \mathrm{W}_\tau & \text{for } X = \{(T\alpha)\cup(N\alpha)\} \end{cases} . \tag{28}$$

If the tool τ is based on model separation, then $\mathrm{G}_\tau \in [0,1]$ quantifies the confidence about the two components of the mixture being successfully separated, and $\mathrm{W}_\tau = 0$. Instead, if τ is not based on model separation, we assign all the mass to the doubt by setting $\mathrm{W}_\tau = 1$, thus yielding the neutral element for Dempster's combination rule [3] and disabling the contribution of this BBA. This phase of the fusion framework is represented in the lower part of Figure 21.

Notice that, for the moment, the above BBA is not linked in any way to that in eq. (27) (they are also defined on different frames, Γ_α and Θ_α respectively). This means that we are not still logically linking local and global information about the presence of the trace.

Trace-based background information. Decision fusion is particularly interesting when tools searching for different traces are used sinergistically. In fact, by knowing the theoretical properties of each forensic trace, in many cases the analyst can explicitly tell whether a combination of traces is plausible or not: this is what we call *trace-based* background information. As it was shown in Section 2, DST allows to express rather easily such information in terms of BBAs, allowing to combine it with the information provided by single forensic tools.

While the general ideas are the same we used for tampering detection, even with regard to trace-based background information some noticeable differences exist when we consider tampering localization. In Section 2 each forensic trace was modeled with one variable, so that only relationships between different traces had to be considered. In the localization scenario, instead, each trace is better represented with two variables (one referring to the local presence of the trace and one to the suitability of the global model). Hence, we must also consider the relationship between these two variables establishing the link between local and global information, and

Table 9: *Example of traces relationships.*

$\mathbf{\Theta}_\alpha$	$\mathbf{\Gamma}_\alpha$	$\mathbf{\Theta}_\beta$	$\mathbf{\Gamma}_\beta$	**Plausible**	**Interpr.**
$t\alpha$	$T\alpha$	$n\beta$	$T\beta$	Y	Tamp.
$n\alpha$	$T\alpha$	$n\beta$	$T\beta$	Y	Auth.
$t\alpha$	$T\alpha$	$t\beta$	$T\beta$	N	-

allowing to change the interpretation of the local output of a forensic tool based on the global information. It is worth noting that the global information about the presence of one trace can also affect the interpretation of different forensic traces. Therefore, we decided to write together these compatibility relationships. A good way to do that in practice is to write a table listing on rows the combinations of variables: each row is then labelled by the analyst as either plausible or not plausible. For plausible rows, the analyst also specifies the interpretation associated to that row in terms of authenticity of the block. Of course, this has to be done only once for a set of forensic traces. An example for two traces α and β is given in Table 9: the first row states that, for any analysis block of an image where the global models of both trace α and β were successfully separated, it is plausible to find only the trace α and not the other; moreover, the interpretation associated to this combination is "the block is tampered with". The second row of the table tells that local absence of both traces is plausible and is to be interpreted as the block being authentic (based on the available information). The last row, instead, states that the two traces cannot be present simultaneously in the same block. The table is truncated for the sake of brevity; the complete version has 16 rows, even though it makes sense to write explicitly only plausible combinations.

Compatibility tables can be easily written in terms of BBA as follows: for a given set $\mathcal{T}$ of traces, let us define as $\Psi = \prod_{j\in\mathcal{T}} \Theta_j \times \Gamma_j$ the common frame of discernment, where $\prod$ and $\times$ denote the Cartesian product. Let us also denote by $\Psi_{\mathrm{PL}} \subseteq \Psi$ the union set of all combinations that are considered plausible. Then, the following BBA declares that combinations that are not plausible must be considered as conflicting information:

$$m^{\Psi}_{\mathrm{rel}}(X) = \begin{cases} 1 & \text{for } X \in \quad \Psi_{\mathrm{PL}} \\ 0 & \text{for } X \notin \quad \Psi_{\mathrm{PL}} \end{cases}. \tag{29}$$

This part of the fusion process is denoted in Figure 21 by the block whose output is m^{Ψ}_{rel}.

Construction of the overall map. By applying Dempster's combination rule to the BBA resulting from traces relationship and those available from single tools, we obtain a single BBA summarizing the available information. Then, it makes sense to compute the belief of the set composed by all plausible combinations whose interpretation is "tampered with", using Definition 2. Notice that this computation must be performed only once for a given set of forensic traces; the resulting formula remains the same for every image, so it can be stored and evaluated when needed. By evaluating the formula for each analysis block of an image, a map taking values in [0,1] is produced, expressing the belief that each is tampered with.

Map refinement by guided filtering. As the vast majority of forgery localization tools process each analysis block independently of the others [9, 23, 29], the resulting localization map are typically affected by noise. In some cases, authors proposed to filter the map to reduce noise (e.g., in [9] median filtering is advised), but this solution could be not sufficient when several maps have to be fused. Moreover, the use of filtering based on a fixed window (as in the vase of median or mean filtering) rises the problem of how to set the window size: a large window produces more reliable results, but reduces the effective resolution of the localization map; conversely, a small window has a better capability to localize forgery (especially in the case of small tampering), but with limited noise reduction capability. To this aim, we propose to exploit what we call *semantic-based* background information, meaning that we let the content of the analysed image

to drive the refinement process. In particular, we follow [30] where the use of a guided filter [31] is described. Guided filter computes the filtered output by considering the content of the guiding image. In this application, the input is the localization map and the guiding image is the image under inspection. The main advantage is that the guided filter transfers the structures of the guiding image to the filtered output (i.e. the filtered map). Moreover, as shown in [31], this filter can be efficiently computed in $O(N)$ time, and this makes it more efficient than other *edge-preserving* filters, as bilateral filter, whose extended version can be found in [32].

4.3 Experimental results

In this section we discuss the experiments that we carried out to prove the validity of the proposed approach.

4.3.1 Case Study

The tools we employ are based on *aligned* double JPEG compression (AJPEG) footprints [9], *non-aligned* double JPEG (NAJPEG) footprints [22] and Color Filter Array (CFA) inconsistencies [29]. We summarize briefly their underlying scenarios.

In [9], it is analyzed a scenario in which an original JPEG image, after some localized forgery, is saved again in JPEG format. Such a forgery disrupts JPEG compression footprints. Examples of this kind of manipulation are a cut and paste from either an uncompressed image or a resized image, or the insertion of computer generated content. In this case, DCT coefficients of unmodified areas undergo a double JPEG compression thus exhibiting double quantization (DQ) artifacts, while, very likely, DCT coefficients of forged areas do not show such artifacts. If the image was not cropped between the first and the second compression, the grid of the DCT coefficients of the first compression is *aligned* to the second one.

In [22] a different scenario is proposed for image splicing. Here, it is assumed that a region from a JPEG image is pasted onto a host image that does not exhibit the same JPEG compression statistics, and that the resulting image is re-compressed in JPEG format. In this case, the forged region exhibits double compression artifacts, whereas the not manipulated region does not. By assuming a random placement of the spliced region, there is a probability of 63/64 that the grid of the DCT coefficients of the first compression is *not aligned* to the second one (NAJPEG artifacts).

In [29], authors propose a forgery localization method based on the traces left by CFA interpolation. The scenario is a one in which a local forgery destroys the correlation introduced by in-camera *demosaicing*. Thus, the forged region does not show CFA artifacts, whereas the remaining part of the image presents them.

4.3.2 Methodology

To simplify our case study, we set the dimension of each block to 8×8 pixels, which represents the minimum resolution on which double JPEG compression based algorithms work. In order to define the mapping from the localization maps to BBAs (Eq. (27)), we adopt the method proposed in [33], choosing the following set of properties to locally characterize the reliability of each tool τ:

1. q_2: the value of the last compression factor, if any;
2. μ: mean value intensity of the block of pixels;
3. σ: standard deviation of the intensity of the block of pixels;
4. q_1: the value of the first compression factor, if any.

It is worth noting that q_1 is not directly observable, but it is estimated by AJPEG and NAJPEG tools, and it is employed only for CFA, since as shown in [29], traces of CFA artifacts could be removed by strong past compression. The generic analysis block is thus described by the vector $v = (o_\tau, q_2, \mu, \sigma, q_1)$, where o_τ denotes the value of the block in the map produced by tool τ (in our case, $\tau \in \mathcal{T} = \{\text{AJPEG}; \text{NAJPEG}; \text{CFA}\}$). By applying the approach proposed in [33], each

vector is associated to scalar values L_τ, N_τ and D_τ (see Eq. 27); as to the parameters required in [33], we used $\alpha = 0.85$ and $\hat{\eta} = 12$ for each tool, whereas $\hat{\gamma} = 0.5$ for CFA tool, $\hat{\gamma} = 512$ for AJPEG tool and $\hat{\gamma} = 2048$ for NAJPEG tool. These values were gathered through 5-fold cross validation and grid search.

Finally, as motivated in section 4.2.1, we define an empirical method to assign values to global variables, telling to what extent the tool successfully separated the two components for its own trace. Since all the considered tools are based on model separation, according to equation (28) we set $W_\tau = 0 \; \forall \tau \in \mathcal{T}$, and we define a linear piecewise function:

$$G_\tau(\rho) = \begin{cases} 1 & \text{for } \rho \geq a \\ \rho/a & \text{for } \rho < a \end{cases}, \tag{30}$$

where the input ρ is the percentage of blocks belonging to the less populated model, as explained in Section 4.2.1. By definition, G_τ takes values in $[0, 1]$ and it also depends on the parameter a, which represents the minimum percentage of blocks allowing a model to be detected. The value of a was derived from experimental evidence, set to $a = 1/8$. The rationale is that two components can be separated if at least $1/8$ of the blocks shows the footprints searched for.

4.3.3 Results

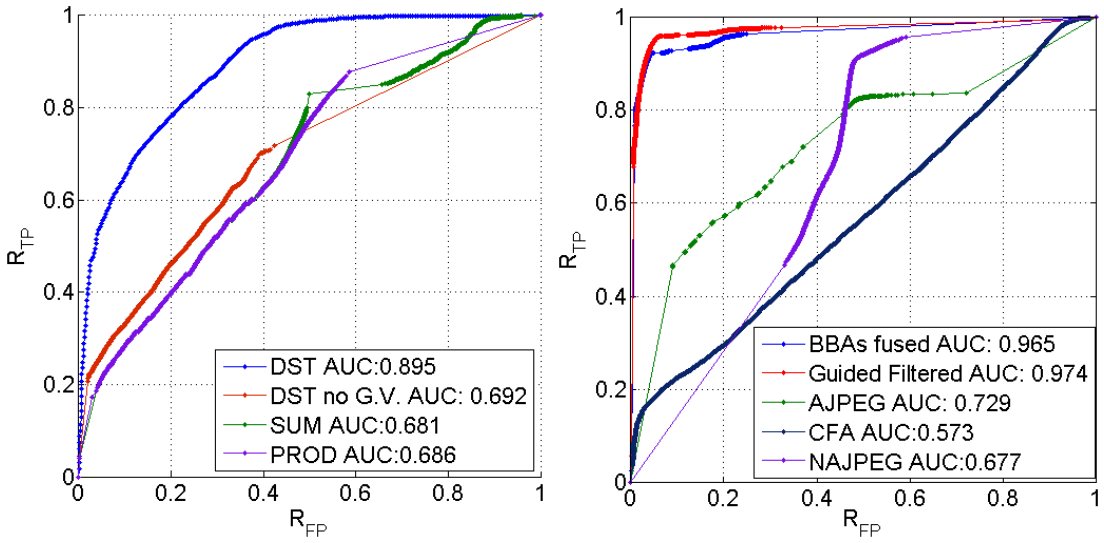
Here we show the improvements in localizing forgeries in an unsupervised scenario. To quantify it, we generate three different sets of images to train and test the proposed framework. Firstly, we define a *training* set to train the BBA mapping module, incorporating *tool-based* background information. The second step is to design a proper dataset (we refer as *testing*) to compare the performance of each tool employed individually with respect to those of the framework. It is worth noting that we assume a *blind* case, i.e. each tool is applied without any a priori information about the type of tampering applied to the image. Finally, we build a dataset of realistic spliced images in order to show the real capabilities of localizing a forged region. The details are listed below.

Training: Starting from 100 uncompressed TIFF images cropped to a 1024×1024 resolution, three different tampering (AJPEG, NAJPEG and CFA destruction) have been applied separately, in such a way that the traces detected by each algorithm have been inserted (or deleted) from the left half of each image. For the AJPEG and NAJPEG traces, the quality factors of the first and second compression are in $\{50, 60, 70, 80, 90, 100\}$, whereas for the CFA footprint, the quality factors employed are in $\{50, 60, 70, 80, 90, 100, \text{Inf}\}$, where Inf represents the case of TIFF uncompressed images. By combining all possible compression factors, we obtain a set composed by 3600 images for AJPEG, 3600 for NAJPEG and 700 for CFA case.

Testing: Starting from 50 uncompressed TIFF images, with a different content from the training set, we apply the same tampering as before to the central block of 512×512 of the images. For AJPEG and NAJPEG traces, the quality factors of the first compression are in $\{60, 70\}$, whereas the quality factors of the second are in $\{80, 90\}$. For the CFA based tampering, a median filtering is applied to remove traces of CFA artifacts. Overall, 750 test images have been created: 200 with AJPEG tampering, 200 with a NAJPEG tampering, 150 with CFA tampering and 200 containing AJPEG and NAJPEG traces at the same time.

Realistic: 19 realistic forgeries have been created through a *cut and past* strategy, by inserting a content (i.e. an object) coming from an image onto another one, and keeping track of the forged region position. The set is composed of 4 TIFF images, whereby an object (without CFA artifacts) is pasted onto another (with CFA artifacts), 6 images with AJPEG footprints , 5 images with NAJPEG footprints and 4 images whereby objects with NAJPEG traces have been inserted in images with AJPEG traces. All forgeries were made in such a way that each footprint is easily detected, since the aim of this dataset is to evaluate the capability of localizing a realistic forgery.

To prove the validity of the framework, we use the *true positive rate* (R_{TP}), measuring the fraction of tampered blocks correctly detected as forgery, and the *false positive rate* (R_{FP}), measuring the fraction of unchanged blocks erroneously detected as forgery. The overall performance of the compared methods are evaluated by plotting its *receiver operating characteristic* (ROC)



(a) (b)

Figure 23: *In Fig. (a), we show a comparison of our framework (blue curve) with the methods proposed in [34], based on the sum (green) and the product (purple) of the output map. Moreover, we show the decrease in the case of absence of global variables (red). The performance are evaluated on the* testing *dataset. In Fig. (b), we show a comparison of the localization capability without post-filtering (blu curve), with the use of guided filtering (red) and the application of each single tool AJPEG (green), NAJPEG (purple) and CFA (black), applied to the* realistic *dataset.*

curve, obtained by thresholding the output maps with a varying threshold value and recording the corresponding values of R_{TP} and R_{FP}. The *area under the curve* (AUC) is finally employed to summarize the discrimination capability of detectors.

The first test is carried out on the *testing* dataset, with the aim to compare our framework to each tool, applied independently and in a blind way. The performance, evaluated in terms of AUC, show that the DST-based framework (AUC = 0.895) outperforms the single detectors AJPEG (AUC = 0.854), NAJPEG (AUC = 0.607) and CFA (AUC = 0.588). It is worth noting that no post-filtering has been applied to the output of fusion step.

As second step, we make a comparison between the performance of our framework and those of the methods proposed in [34], based on the sum and the product of the output map provided by each tool. Moreover, the performance of the framework are evaluated by employing or not global variables, as defined in Section 4.2.1. The results are shown in Fig. 23 (a) by means of ROC curves, evaluated on the *testing* dataset. As we can see, the proposed framework has the best capability of localizing forgeries, and the introduction of global variables dramatically impacts the performance. This is explained by the fact that the introduction of global variables provides further information about the reliability of the value given by a tool. Finally, we present the localization capability of the framework in the case of realistic tampering. In Figure 23 (b), we show the performance of the method without post-filtering and in case of guided filtering at the end of the fusion framework. Moreover, a comparison with each tool performance is proposed. As expected, the refinement by using guided filtering increases the accuracy in localizing realistic forgeries. Even in this case, the DST-based framework has better capabilities with respect to each single tool, applied independently and in a blind way.

5 Countering counter-forensics via multi-clue analysis

As a new challenge to Image Forensics, anti-forensic (AF) methods are emerging, whose goal is to remove the footprints left during processing, making forensic analysis harder [35]. On the other hand, AF tools may leave their own footprints, and counter-anti-forensic (CAF) tools are being designed to detect them as well. In such a scenario, the forensic analyst needs to simultaneously tackle with both the variety of detectable footprints and the presence of an adversary equipped with AF tools. If CAF tools are available to the analyst, a more robust analysis can be carried out, provided that outputs from IF tools and their CAF versions are interpreted properly, thus going back to multi-clue analysis.

For this reason, we started investigating the possibility offered by the adoption of data fusion framework in a CAF scenario. Also in this case, one could opt for the intuitive solution of using all the available tools in an additive fashion ("OR" fusion rule), that means classifying the image as tampered when either an IF or a CAF tool detects the footprint it is looking for. This approach, however, does not take into account the following facts: i) as usual classical forensic tools may be searching for mutually exclusive traces, so some combinations of tool outputs could be excluded; ii) for a given footprint, IF and CAF algorithms are expected to be in contradiction, so if both kinds of tools detect the footprint they look for this should at least raise some doubts about the correctness of the outputs; and iii) detecting some kinds of anti-forensic processing does not necessarily imply that the image is a fake (e.g., full-frame linear filtering may be seen as an AF technique, but usually it has to be considered a common, innocent operation).

In this section, we describe a first attempt to use multi-clue analysis as a counter-forensic tool and some repliminary results that we obtained by applying the proposed framework to a specific case-study.

5.1 Integration of CAF methods into the DST fusion framework

We now investigate how CAF tools can be integrated in the DST fusion framework described in the previous sections. The main idea behind CAF is to search for the traces that are left by anti-forensics tools; indeed, most existing AF tools only attempt to remove some kind of forensic trace (e.g., comb-shaped histograms in the pixel or DCT domain), but they do not care about making the statistics of the produced signal close to that of an untouched content. Only very recently the first steps have been taken towards universal counter-forensic techniques, but they are limited to the specific scenario where analyst's tools are based on first-order statistics [36]. As a result, we can say that the application of common AF tools may introduce new footprints that are possibly harder to detect, but still detectable.

In such a scenario, the forensic analyst should consider the possibility that the analyzed content has been manipulated and then *cleaned'* by the forger. This means that, whenever they are available, CAF tools should also be part of the set of tools employed during the analysis. If we go back to the fusion framework described earlier, CAF tools can be modeled as standard IF tools: they simply search for a specific trace and provide some information about its presence or absence. However, the complementary nature of these tools compared to IF tools raises some questions about how they should be integrated within the fusion framework. We can distinguish between two possible approaches:

- a *cascade scheme*, where outputs from IF tools are first merged using the system in Fig. 1, and CAF tools are considered later, possibly introducing knowledge about relationship between CAF and IF traces;
- a *mixed scheme*, where IF and CAF tools are treated at the same level.

The cascade scheme is more convenient from a complexity point of view: after fusing information about IF traces, the analyst may marginalize such information to reduce the cardinality of sets. For example, the analyst may summarize the information about all the traces of a specific family (e.g., JPEG-related traces) into a single variable, discriminating between presence or absence of that family of traces. Then, the compacted belief structure could be merged with the information coming from CAF tools. The downside of the cascaded structure is the possible

over-simplification induced by grouping traces into families. Indeed, relationships between traces searched by IF and CAF tools are not trivial: in the simplest case, by applying an AF tool, the adversary erases the trace searched by a IF tool while introducing the trace searched by a CAF tool, so that these two traces are simply mutually exclusive. However, as we will see in the next sections, it may be that due to the application of AF tools the relationships between the standard IF traces are changed: traces that could not coexist in a *standard'* forgery scenario, may become compatible due to the application of AF tools and viceversa. This can happen even for traces within the same family, so that the marginalization would cause an over-simplification of the problem, since it would impede to correctly update relationships in presence of CAF traces. Finally, we can state that it is safer to treat IF and CAF traces at the same level, and the higher complexity seems an unavoidable price to pay. Moreover, cheaper solutions are not easy to find: for example, the above facts have a heavy impact also on fusion frameworks based on machine-learning. The increasing number of possible combinations of IF and CAF traces makes it hard to devise proper training sets without generating and analyzing an exponential number of training samples.

It is worth remarking that the application of an AF tool targeted to remove a particular forensic trace may affect also the detectability of other traces; surprisingly, the effect could also be to *increase* the detectability of other traces. Even more, application of a targeted AF tool may introduce an IF trace that was not present beforehand. This means that, by using a multi-clue analysis, the analyst may be able to counter the effect of AF tools without employing CAF tools, and simply relying on the complementary capabilities of the IF tools included in the framework (we will see an example of this effect later on).

Interpretation of AF Traces. A controversial point about searching anti-forensics traces is that some processing operators have a double valence: they can both be used "benignly" to enhance the quality of the image, or they can be used "maliciously" as a tool for erasing traces of previous processing. For instance, Kirchner et al. showed how median filtering can be used to erase traces of resampling [37], that are often used to localize zoomed or rotated areas within a tampered digital image. Due to this fact, the forensic analyst should try to detect application of processing operators that affect traces searched by IF tools. On the other hand, it could seem hasty to classify an image as manipulated when only traces of possibly benign processing operators are detected: although benign processing can be considered as a form of manipulation, we believe it is of interest for the analyst to discriminate between global processing, like filtering, and forging operations like splicing. A solution to solve this apparent deadlock is to deepen the analysis of AF traces moving, when it is possible, from a global perspective (i.e., search for the trace over the whole image) to a local perspective (search the trace separately on the suspect region and on the rest of the image). Indeed, while revealing a global application of a processing operator may be considered acceptable, detecting *inconsistent* presence of traces among different regions is much more alarming. Finally, what we are facing with is an interpretation problem: should the presence of traces characterizing "benign" processing operators raise integrity warnings? Should they do that only when they are present in an inconsistent way across the same image? We believe the answer to these questions depends on the field of application, and should be left to the analyst. Therefore, the fusion framework should not force any interpretation, while enabling the possibility for the analyst to select the preferred one.

The above arguments can be included in the DST fusion framework in a rather intuitive way. When introducing information about the presence of ambivalent processing operators, we introduce *two* traces within the framework: one concerning the presence of operator traces within the suspect region, and one concerning the presence of operator traces within the rest of the image. For example, suppose we have a tool G searching for a CAF trace γ: we propose to use tool G to assign BBAs to the frame of discernment associated to the set $\Theta_\gamma^I = \{t\gamma^I, n\gamma^I\}$ and $\Theta_\gamma^O = \{t\gamma^O, n\gamma^O\}$, where $t\gamma^I$ and $n\gamma^I$ denote, respectively, presence or absence of the trace inside the analysed region, and $t\gamma^O$ and $n\gamma^O$ denote presence or absence of the trace outside the analysed region. If we consider the product set $\Theta_\gamma^I \times \Theta_\gamma^O$, we obtain all possible combinations of presence and absence of the CAF trace inside and outside the analysed region. In this way,

the analyst can choose, for example, whether presence of the trace both inside and outside the region, that is the event $(t\gamma^I, t\gamma^O)$, should be considered as an integrity violation or not.

5.2 Two case studies

We now apply the ideas outlined above to two case studies whose goal is to evaluate the practical impact of CAF tools when the adversary is equipped with AF technologies. We focus on a widely studied task in image forensics, that is splicing detection: given a digital image, splicing detection aims at understanding whether a suspect region (either specified by the analyst or automatically selected in some way) has been pasted from another image.

In devising the two case studies, we adopted an experimental setup which is very similar to the one used for detecting splice detection by relying on double-JPEG compression traces. For sake of clarity, we report here the main assumptions behind the adopted setting.

The analyst employs five different tools for splicing detection based on the analysis of three different JPEG-related traces:

- the tool by Bianchi et al. [9] and the tool by Luo et al. [21] searching for traces of not-aligned double JPEG compression (JPNA);
- the tool by Bianchi et al. [22] and the tool by Lin et al. [23] searching for traces of aligned double JPEG compression (this trace will be called JPDQ from now on);
- the tool by Farid [24] searching for traces of the so-called "JPEG-ghost" (JPGH).

The compatibility relationship between these traces is reported in Table 1.

Comb. num	**JPNA**	**JPDQ**	**JPGH**	**Interpr.**
1	0	0	0	Non-tampered
2	0	0	1	Tampered
3	0	1	0	-
4	0	1	1	Tampered
5	1	0	0	Tampered
6	1	0	1	-
7	1	1	0	-
8	1	1	1	Tampered

Table 10: *Trace relationship table: each row forms a combination of presence (1) and absence (0) of traces. In the rightmost column we see the interpretation of each combination, where impossible combinations are denoted by a dash. Notice that only 5 out of 8 combinations are possible.*

Switching to the adversary's point of view, regardless of counter-forensic strategies, four different cut-&-paste procedures are considered to create a splicing starting from two images (at least one of which is in JPEG format), that are described in Table 3. As the reader can see from the table, different procedures introduce different combinations of IF traces.

By relying on these ideas, we upgrade both the forger's skills by introducing counter-forensic methods, and the analyst's skills by providing proper CAF tools. The two considered case studies differ in that, in the first one, the forger wants to obtain a spliced JPEG image where traces of double encoding are concealed, while in the second (more complex) case the product of the splicing must show no traces of compression at all (thus erasing all traces that may be used by JPEG-based IF tools).

Class	Procedure	Traces in inner region	Traces in outer region
Class 1	Region is cut from a JPEG image and pasted, breaking the 8x8 grid, into an uncompressed one; the result is saved as JPEG.	JPNA	-
Class 2	Region is taken from an uncompressed image and pasted into a JPEG one; the result is saved as JPEG.	-	JPDQ JPGH
Class 3	Region is cut from a JPEG image and pasted into an uncompressed one in a position multiple of the 8x8 grid; result is saved as JPEG.	JPGH	-
Class 4	Region is cut from a JPEG image and pasted (without respecting the original 8x8 grid) into a JPEG image; the result is saved as JPEG	JPNA	JPDQ JPGH

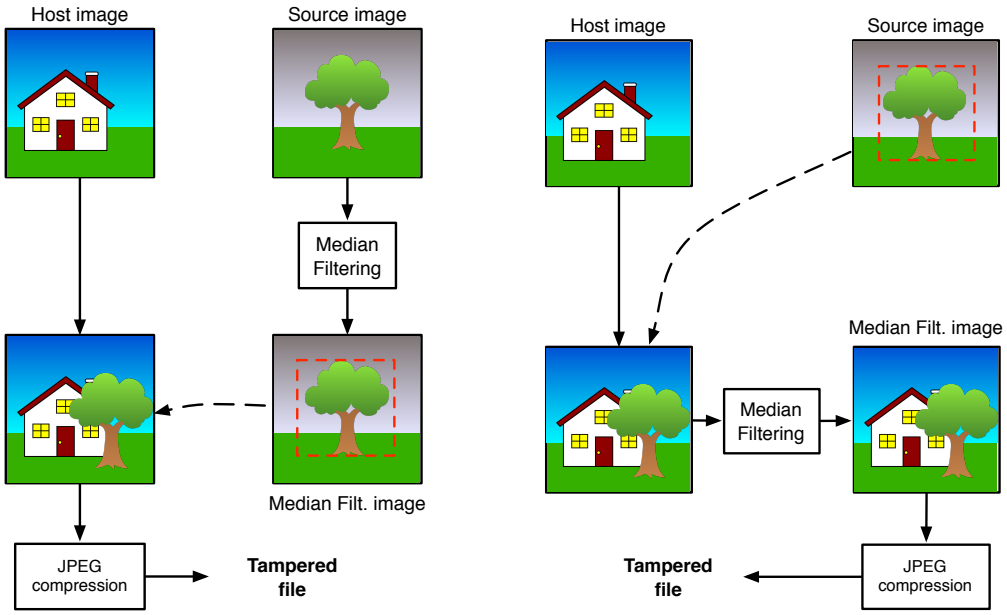
Table 11: *Procedure for the creation of different classes of tampering in the training dataset.*

5.2.1 Splicing Detection in the Presence of Double Encoding Concealment

In this case study we consider a forger who wants to generate a splicing starting from two images, at least one of which is in JPEG format, and finally encode the result as a JPEG image. Since it is known that many IF tools exist based on the analysis of double JPEG compression traces, the adversary wants to conceal these traces. A simple but effective way to accomplish such a task is to perform a median filtering between the two compressions, thus avoiding pixels from showing traces of double quantization. Although other kinds of filtering operators could also be used, we believe median filtering is preferable because of its non-linear nature, that complicates the task of the analyst.

The adversary can apply the filter either before of after the cut-&-paste operation, resulting in two different scenarios (Figure 24). Applying MF to the whole image (right figure) gives more chance of disrupting traces but also increases the probability of being detected by CAF algorithms. Since median filtering is known to be a good AF method [38], we can reasonably think that the analyst adopts proper counter-measures. Therefore, we include the tool for detecting MF in JPEG compressed images proposed by Kirchner et al. [38] within the analyst's pool of tools. This CAF tool uses SPAM (subtractive pixel adjacency matrix) features [39] to characterize the relationships between neighbouring pixels, and classify the analysed region as median filtered or not using machine learning techniques. We trained the classifier following indications in the original work [38].

After choosing the set of IF and CAF traces to search for, the analyst needs to reason about the relationships between these traces. First of all we notice that median filtering is a "forensically ambiguous" operator because it can be used both benignly to remove noise from the image and maliciously, as an AF tool. Based on our previous discussion, we find it appropriate to use the CAF tool to analyse *separately* the suspect region and the rest of the image, so to distinguish



(a) (b)

Figure 24: *Two possible methods available to the analyst to conceal forensic traces that are left during tampering using median filtering.*

Comb.	JPNA	JPDQ	JPGH	CAF-IN	CAF-OUT	Interpr.
1	0	0	0	0	0	Non-Tampered
2	0	0	0	0	1	Tampered
3	0	0	0	1	0	Tampered
4	0	0	0	1	1	Non-Tampered
5	0	0	1	0	0	Tampered
6	0	0	1	1	0	Tampered
7	0	0	1	1	1	Tampered
8	0	1	1	0	0	Tampered
9	0	1	1	1	0	Tampered
10	0	1	1	1	1	Tampered
11	1	0	0	0	0	Tampered
12	1	0	0	1	1	Tampered
13	1	0	1	0	1	Tampered
14	1	1	1	0	0	Tampered
15	1	1	1	1	1	Tampered

Table 12: *Compatibility relationships for IF and CAF traces considered in the first case study. Each row considers a combination of presence (1s) or absence (0s) of considered traces. Notice that the presence of the CAF trace is treated separately for the suspect region (CAF-IN column) and the rest of the image (CAF-OUT). Only plausible combinations are reported in the table, those combinations that are not listed are theoretically incompatible.*

between a malicious and innocent use of some processing.

We can finally write relationships, updating Table 1 so to account for the presence of traces of median filtering (both inside and outside the analysed region): the result is plotted in Table 12. Notice that, for brevity, only plausible combinations are reported. First of all, notice that combination number 4 reads as follows: when traces of median filtering are detected throughout the whole image, and this is the *only* detected trace, then the image is considered non-tampered. This is the chosen interpretation in this case study; however, nothing prevents the analyst from changing this interpretation, for example to account for a specific setting where any retouch of the image is not acceptable. Moving to the rest of the table, most of the entries are rather intuitive: integrity violation is detected when at least one of the IF tools finds the trace it is looking for, and also when traces of MF are present in only one part of the image (because inconsistent use of filtering is interpreted as a malicious behaviour). The only combination leading to interpret the image as "non-tampered" is the one where none of the traces are present. On the other hand, some interesting combinations exist. Let us consider the combination number 13: if we focus only on IF traces (columns 2-4), we recognize one of the impossible combinations according to Table 1, namely number 6. Yet, if we take into account CAF traces (columns 5-6), the combination becomes possible: indeed, median filtering applied to the external region makes it impossible to detect traces of double quantization (JPDQ) there, while the other two traces are still detectable because the inner region was not affected by filtering. This is a very good example supporting the mixed architecture despite its heavier cost compared to the cascaded scheme.

Experimental Results. Given the set of tools defined in this case study, and the compatibility relationships in Table 12, we want to investigate the performance of the multi-clue framework. To do that, we started from 100 uncompressed and heterogeneous TIFF images (indoor, outdoor, landscapes, etc.) of size 1024×1024 pixels, and we generated a dataset of 8000 images, of which:

- 2000 untouched images, obtained by simply applying JPEG compression;

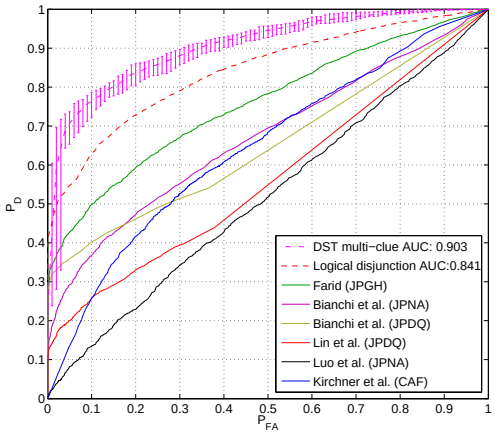


Figure 25: *ROC curves obtained on the whole dataset by each of the considered IF and CAF tools (solid lines), and by the two decision fusion methods (dashed lines). The ROC curve relative to the DST-based method shows the maximum and minimum value obtained through all the train-test iterations.*

- 500×4 spliced images without AF, generated using the 4 different cut&paste attacks reported in Table 3;
- 500×4 spliced images, to which AF is applied to the spliced region only, according to Figure 24(a).
- 500×4 spliced images, to which AF is applied to the whole image, according to Figure 24(b).

The spliced region has always a size of 256×256 pixels, and is located in the center of the image. Possible values for the quantization quality of the first compression, denoted with Q_1, were chosen so that $Q_1 \in \{40, 45, ..., 80\}$, while the second quality factor Q_2 was defined as $Q_2 = Q_1 + \delta$, with δ randomly chosen from $\{+5, +10, +15, +20\}$.[7]

We run the 5 IF tools and the CAF tool for median filtering detection on all images, then we employed the proposed method to calculate merged mass assignments. Since BBA mapping requires training information, 80% of the images in the dataset (selected at random) were used to train each tool separately, and the rest were used for testing the system. This procedure was repeated 10 times to increase the statistical significance of the results.

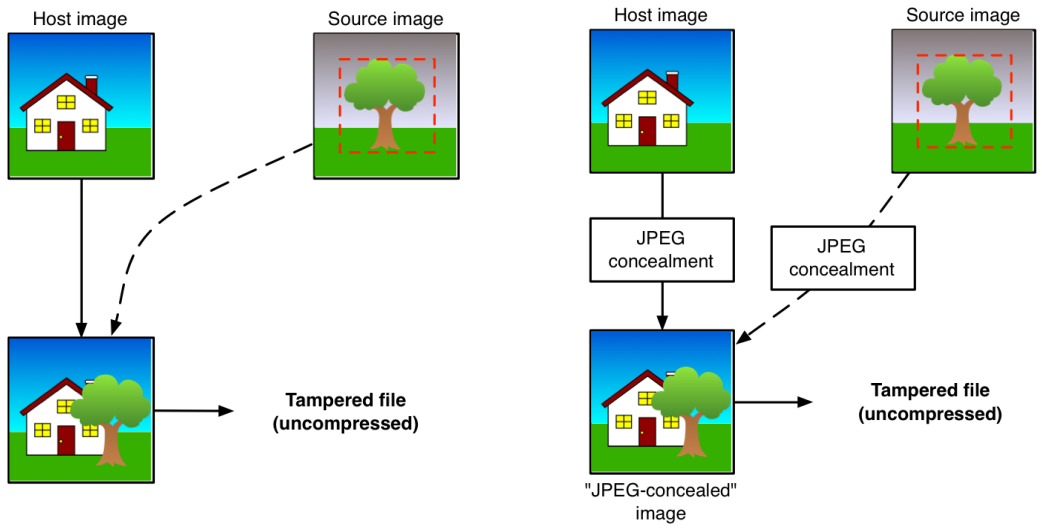
For each image, we evaluated the belief for the set containing all combinations of Table 12 whose interpretation is "tampered". Then, the obtained belief was thresholded so to obtain the final decision. Figure 25 shows the Receiver Operating Characteristic (ROC) curves obtained with: the proposed method (dash-dot curve), with every single IF and CAF algorithm alone (solid curves), and also with a simple decision fusion rule (dashed curve), that is logical disjunction (i.e., the image is classified as tampered if at least one algorithm detects a trace). The most evident fact is that decision fusion strongly helps the analyst in the presence of an adversary: this confirms that AF methods are less effective when the analyst uses a pool of complementary IF tools. We also see that the proposed system outperforms the logical-disjunction method, a rather trivial yet widely used approach.

5.2.2 Splicing Detection in the Presence of JPEG Coding Concealment

In this scenario, the forger starts with two images (at least one JPEG coded) and wants to produce a splicing that is not JPEG compressed. In order to accomplish this task, the forger

[7] We did not consider the case where the second compression is at a lower quality than the former (that is, $Q_1 < Q_2$) because it is known that JPEG-based tools do not perform well in such a setting.

can either adopt a naive approach (Figure 26(a)) where the spliced image is just stored in an uncompressed format, or a smarter approach where an AF tool is applied to conceal traces of previous JPEG compression (Figure 26(b)) before creating the splicing. A good candidate for this AF task is the tool proposed by Stamm et al. [40], that removes the characteristic trace left by JPEG compression, namely gaps in the histograms of DCT coefficients. Gaps are filled by adding a dithering noise to coefficients so to make their distribution resemble that of an uncompressed image. Of course, the forger applies this AF method only to pixels coming from JPEG-coded images.



(a) (b)

Figure 26: *Two possible methods available to the analyst to produce an uncompressed spliced image. In figure (b), JPEG concealment is applied only to pixels coming from JPEG-compressed images.*

Let us now consider the analyst's side. The IF tools we considered are all based on JPEG artifacts, and we may think that the analyst is stuck when an uncompressed image has to be analyzed. Yet, the analyst may consider the following possibilities:

1. the image is actually non-tampered;
2. a splicing has been generated starting from two uncompressed images;
3. a splicing has been generated starting from one (or both) JPEG-coded images, and it has been stored with no compression;
4. a splicing has been generated starting from one (or both) JPEG-coded images, traces of JPEG compression have been removed using an AF tool, finally the result has been stored with no compression;

With the available set of IF tools, the analyst is not able to detect forgeries that are performed according to point 2. Still the scenario in point 3 can be handled with a rather simple approach: by recompressing images, and searching for traces of aligned or not-aligned double compression. If such traces are found, and they are inconsistent between the suspect region and the rest of the image, then the splicing is properly exposed. However, a clever forger would probably have concealed traces of JPEG compression, like in Figure 26(b). This is where we upgrade the tools of the analyst, introducing the CAF tool [41], proposed by Valenzise et al., that allows to detect

Comb.	JPNA	JPDQ	JPGH	CAF-IN	CAF-OUT	Interpr.
1	0	0	0	0	0	Non-Tampered
2	0	0	0	0	1	Tampered
3	0	0	0	1	0	Tampered
4	0	0	0	1	1	Tampered
5	0	0	1	0	0	Tampered
6	0	0	1	0	1	Tampered
7	0	0	1	1	0	Tampered
8	0	0	1	1	1	Tampered
9	0	1	1	0	0	Tampered
10	0	1	1	1	0	Tampered
11	1	0	0	0	0	Tampered
12	1	0	0	0	1	Tampered
13	1	0	1	0	1	Tampered
14	1	1	1	0	0	Tampered

Table 13: *Compatibility relationships for IF and CAF traces considered in the second case study. Each row considers a combination of presence (1s) or absence (0s) of considered traces. Only plausible combinations are reported in the table, those combinations that are not listed are theoretically incompatible.*

JPEG compression even in presence of the previously mentioned AF attack. By using this CAF tool, the analyst can expose traces of previous JPEG-compression and, most interestingly, search for inconsistent traces inside and outside the suspect region.

We stress again that, in practice, the analyst does not know which processing chain the suspect image underwent. A good strategy, therefore, is to run the available IF and CAF algorithms and properly interpret their outputs. To this end, in this case study we let the analyst perform the following actions:

- when the image is in JPEG format: apply the IF tools, and disable the CAF tool;
- when the image is uncompressed, do both the following:
  - use the CAF tool to expose possible traces of previous JPEG compressions;
  - perform a high-quality JPEG compression and run the IF tools.

In order to maximize the benefits from this joint analysis, the analyst must then provide knowledge about IF and CAF traces relationships, together with the interpretation. Table 13 shows a possible, reasonable, choice for the considered case study. Also in this case, the mixed scheme allows to properly account for non-trivial relationships. For example, let us consider line 13: the combination of IF traces would normally not be possible (see combination 6 in Table 1), but it becomes possible when JPEG traces were removed from outside of the suspect region, and not inside it. Considering Figure 26(b), this would actually happen when the host image was JPEG compressed and the source image was not. Finally, we point out that, in this case study, presence of AF traces is interpreted as a manipulation even when no IF traces are found (row 4 of Table 13): this is is motivated by the fact that hiding traces of JPEG compression can never be considered a "benign" processing, since it does not bring any positive effect to the image.

Experimental results. Based on the above description, we generated a dataset of authentic and forged images so to evaluate the performance of IF tools and of the multi-clue analysis system. We generated:

- 600 untouched images with no compression (TIFF format);

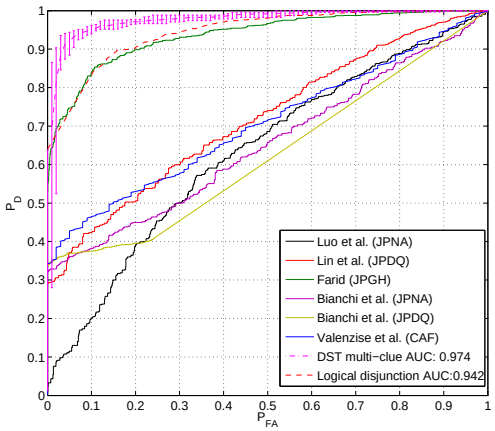


Figure 27: *ROC curves obtained on the second case study dataset by each of the considered IF and CAF tools (solid lines), and by the two decision fusion methods (dashed lines). The ROC curve relative to the DST-based method shows the maximum and minimum value obtained through all the train-test iterations.*

- 600 untouched and JPEG compressed images;
- 100×4 spliced images without AF, generated using the 4 different cut&paste attacks reported in Table 3;
- 100×4 spliced images, that are simply stored after the cut-&-paste, without compression, according to Figure 26(a);
- 100×4 spliced images, to which AF is applied to remove traces of JPEG compression, according to Figure 26(b).

We used the same parameters (tampering size, compression strengths, etc.) that we used before. Uncompressed images were analysed both using the CAF tool [41] and the JPEG-based tools, after compressing them with quality 95.[8] On the other hand, the CAF tool was not run on JPEG-compressed images, mapping its output to a vacuous belief assignment. We also maintained the same partitioning between training and test samples for evaluating the DST based multi-clue framework. Results obtained by single tools and fusion techniques are plotted in Figure 27, and confirm that multi-clue analysis allows the analyst to effectively counter the presence of AF techniques. Interestingly, the tool based on JPEG ghost [24] yields good performance notwithstanding the presence of JPEG anti-forensics. We actually found that, even after application of the employed AF tool [40], the algorithm by Farid was still able to detect the pasted region in some cases. This is probably due to the way the tool works, namely accumulating in the spatial domain the contribution coming from all DCT coefficients, without explicitly modelling their histograms.

[8]It has been shown [33] that the best case for JPEG-based forensic analysis is when the last encoding was at high quality, but not the highest possible.

6 A theoretical framework for multi-clue forensics analysis under adversarial conditions

As we have already shown in the previous section, multi-clue forensics analysis plays a major role to combat counter-forensics techniques. In fact, it is arguable that it would be very difficult, if not impossible, for an attacker to delete all the traces the forensic analyst may rely on. The few works on multi clue forensics analysis in adversarial setting published so far, are rather heuristic and focus only on specific problems, thus making it difficult to understand the ultimate potentiality of multi-clue analysis as a CAF tool.

In this section, we present a first attempt to fill this gap. Specifically, we describe a theoretical framework which models the interplay between the forensic analyst and the attacker in a multiple-clue (or multiple-object) forensics scenario. Specifically, the proposed framework analyzes a general Hypothesis Testing (HT) problem assuming that the HT can be carried out by relying on multiple sources of information. In fact, this is quite a common situation in multimedia forensics. For instance, the HT may correspond to identify the source of an image or a video by relying on the information provided by a number of forensic tools, each analyzing a different aspect of the image/video. In the case of still images, for instance, the tools may analyze different color bands, or different frequency coefficients, in the case of video, the observables may refer to the audio and video tracks and so on.

Even if the original target of our analysis was giving a theoretical background to adversarial multimedia forensics, the horizon of the developed framework is wider and encompasses applications like biometrics, digital watermarking, steganography and steganalysis, network intrusion detection, traffic monitoring, cognitive radio and many others.

In the next subsections, we first give a brief introduction to adversarial signal processing [1]. Then we present our theoretical framework for multi-clue (referred to as multiple-object in the rest of the section) hypothesis testing. In doing so, we will refer to previous works on adversarial source identification [2, 42]. Moreover, we will privilege the clarity of exposition with respect to mathematical rigor, so in some cases we will only state theorems without giving a formal proof.

6.1 Adversarial Hypthesis Testing

Adversarial Signal Processing (Adv-SP), i.e. the study of signal processing techniques explicitly thought to withstand the attacks of one or more adversaries aiming at system failure, is receiving an increasing attention due to its applicability in a wide number of scenarios, including multimedia forensics, biometrics, digital watermarking, steganography and steganalysis, network intrusion detection, traffic monitoring, video-surveillance, just to mention a few [1]. Adversary-aware hypothesis testing (or binary decision) is undoubtedly one of the most common problems in Adv-SP, due to its importance in several application scenarios. In multimedia forensics, for instance, the analyst has to decide whether a document has been generated by a given source (a specific camera or a camera model), or has undergone a given processing. In spam filtering, e-mail messages have to be classified either as spam or authentic messages. In 1-bit watermarking, the detector has to decide whether a document is watermarked or not, while it is the goal of steganalysis to distinguish between cover and stego-images. In yet other situations, the security of a system relies on the capability of distinguishing malevolent users from fair ones (again a binary classification problem). Even if specific solutions have been proposed for each of the above problems, the need for a general theoretical framework that models the interplay between the analyst and the attacker is becoming evident. In [2, 42], a game-theoretic framework has been introduced to analyze the hypothesis testing problem under adversarial conditions. By assuming that the analyst can rely only on first order statistics of the observables and that the attacker has to satisfy a distortion constraint, the asymptotic equilibrium point of the game is derived when the length of the observed sequence tends to infinity.

Within AMULET, we have extend the framework introduced in [2] to deal with binary hypothesis testing under multiple observations (multi-clue analysis). In addition to multimedia forensics, this is a relevant scenarios in several applications, data fusion, distributed hypothesis

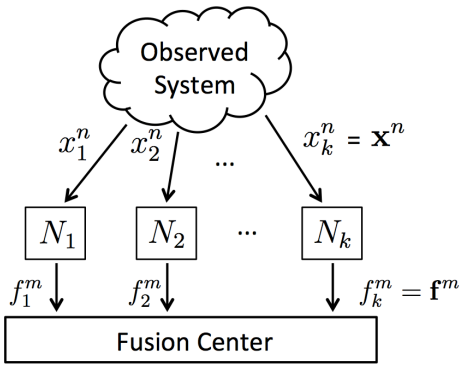


Figure 28: *The multiple-observation hypothesis testing setup.*

testing and detection [43], sensor networks [44] and cognitive radio networks [45].

In all these cases, a fusion center must take a binary decision about the status of a system by relying on a number of observations made available by different sensors (as in [43]) or a number of traces detected by different investigation tools as in multi-clue forensics analysis. In many situations, it is possible that an attacker (or more attackers) corrupts the observations or deliberately provide misleading data to induce a decision error at the fusion center. It was the goal of our research to build a general information-theoretic framework to analyze the above situations and devise the optimal strategies for both the analyst and the attacker in a game-theoretic sense, that is by determining the equilibrium point of the game.

6.2 The setup

A sketch of the Multiple-Observation Hypothesis Testing (MO-HT) that we have consider in our research is reported in Fig. 28. The status of a system is observed by k nodes which gather k observation sequences, $x_1^n, x_2^n \dots x_k^n$, each of which consists of n samples, i.e., $x_l^n = (x_{l,1}, x_{l,2} \dots x_{l,n}), l = 1 \dots k$. The nodes summarize their observations into k feature sequences of length m ($m \leq n$), $f_1^m, f_2^m \dots f_k^m$, with $f_l^m = (f_{l,1}, f_{l,2} \dots f_{l,m}), l = 1 \dots k$. The summaries are sent to a fusion center which has to either accept or reject a certain hypothesis H_0 about the status of the system. This is a very general setup that can be used to model a wide variety of situations. As an example, the nodes may be part of a sensor network and the observed sequences $x_1^n \dots x_k^n$ may describe the physical state of the system over time. e.g., the temperature, measured at different locations. As to the summaries, in the simplest case they coincide with the observed sequences. More often, they are obtained by extracting a number of features from the observed sequences, or by taking a local decision on the system status. In the latter case, $m = 1$ and $f_l^m = 0$ or 1 depending on the local decision on the validity of hypothesis H_0 taken by node l.

In the multimedia forensics case, the observed system is a document, for instance an image or a video, which is analyzed by means of different tools (identified here by $N_1, N_2 \dots N_k$). Each tool analyzes a different aspect of the document. In the case of still images, for instance, the tools may analyze different color bands, or different frequency coefficients, in the case of video, the observables may refer to the audio and video tracks and so on. The tools extract a number of features and send them to a data fusion center, that is in charge of taking the final decision on a certain aspect of the analyzed document (e.g. its origin). As in the distributed hypothesis testing scenario, two extreme cases are obtained when the features correspond to the entire set of observables, and when each tool takes a local decision and the fusion is carried out at the decision level.

When MO-HT is framed in an adversarial setting, we must take into account the possibility that an adversary corrupts part of the system so to induce a decision error. We consider two main possibilities. In a first case, we assume that the attacker corrupts h out of k summaries. This is possible if the attacker seizes h nodes or if he controls h links between the nodes and the fusion center. Two sub-cases are possible depending on whether the attacker can choose which nodes he is going to attack or not. For the rest, we do not put any further limitations on the attacker's

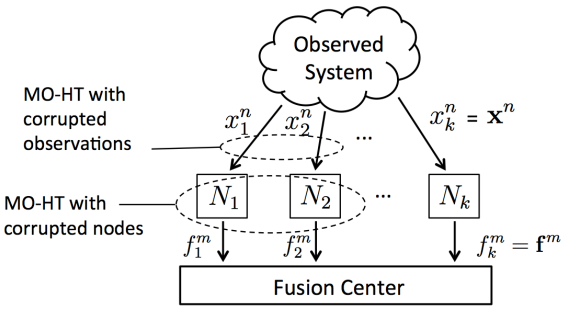


Figure 29: *Multiple-observation hypothesis testing under adversarial conditions.*

actions. In the following, we will refer to this setting as MO-HT with (chosen) corrupted nodes[9].

In a second scenario, the nodes and the links between the nodes and the fusion center are under the full control of the analyst and hence the attacker can only modify h out of k observed sequences. This is typically the case in multimedia forensics analysis, in which an analyst studies various aspects of the document at hand, and takes a decision on the provenance or integrity of the document by fusing the results of the different analyzes. The attacker, on his side, modifies the document so to hide its true origin or its previous history. In these cases, it makes sense to require that the amount of modification the attacker can introduce into the document is limited. In the following, we will refer to this scenario as MO-HT with corrupted observations. A graphical representation of the two scenarios considered in the paper is given in figure 29.

Several versions of the two general settings described above are obtained depending on the actions allowed to the attacker and the analyst, their specific goals, the knowledge they have about the system, including its status and its statistical characterization, the knowledge that the attacker has on the links and nodes that he does not control and so on. In the next sections, we will analyze some of these variants, by framing them into a rigorous game-theoretic setting. As we will see, game-theory provides a natural and flexible way to take into account all the above information and to study the optimal strategy of the two players in terms of game equilibrium and achievable payoff.

6.3 Dominant fusion strategies for the defender

As anticipated, we use game-theory to give a formal definition of the MO-HT problems outlined in the previous section. In this section we adopt the perspective of the analyst, hereafter referred to as the Defender (D), defining his goals, his possible actions and deriving the optimum fusion strategies under some general assumptions.

6.3.1 Game-theory in a nutshell.

A 2-player game can be seen as a 4-uple $G(\mathcal{S}_1, \mathcal{S}_2, u_1, u_2)$, where $\mathcal{S}_1 = \{s_{1,1} \dots s_{1,n_1}\}$ and $\mathcal{S}_2 = \{s_{2,1} \dots s_{2,n_2}\}$ are the set of strategies the first and the second player can choose from, and $u_l(s_{1,i}, s_{2,j}), l = 1, 2$, is the payoff of the game for player l, when the first player chooses the strategy $s_{1,i}$ and the second chooses $s_{2,j}$. A pair of strategies $(s_{1,i}, s_{2,j})$ is called a profile. When $u_1(s_{s1,i}, s_{2,j}) + u_2(s_{1,i}, s_{2,j}) = 0$, the win of a player is equal to the loss of the other and the game is said to be a zero-sum game. In the set-up adopted in this paper, $\mathcal{S}_1$, $\mathcal{S}_2$ and the payoff functions are assumed to be known to the two players. In addition, we assume that the players choose their strategies before starting the game without knowing the strategy chosen by the other player (strategic game).

A common goal in game theory is to determine the existence of equilibrium points, i.e. profiles that, in *some sense* represent a *satisfactory* choice for both players [46]. The most famous

[9]In principle we should distinguish between an adversary that takes full control of the nodes and an adversary that controls only the links between the nodes and the fusion center, since in the former case the attacker can observe the sequences x_l^n of the corrupted nodes, thus acquiring information about the system status. In this paper we consider an omniscient attacker, hence making the distinction between the two cases irrelevant.

equilibrium notion is due to Nash. Intuitively, a profile is a Nash equilibrium if each player does not have any interest in changing its choice assuming the other does not change its strategy. Despite its popularity, the practical meaning of Nash equilibrium is doubtful, since there is no guarantee that the players will end up playing at the equilibrium. A notion with a more practical meaning is that of dominant equilibrium. A strategy is said to be strictly dominant for one player if it is the best strategy for the player, no matter how the other player decides to play. In many cases dominant strategies do not exist, however when one such strategy exists for one of the players, he will surely adopt it (at least under the assumption of rational behavior). The other player, in turn, will choose his strategy anticipating that the first player will play the dominant strategy. It is then easy to see that when a dominant strategy exists, the players have only one rational choice called the only rationalizable equilibrium of the game [47]. Games with the above property are called *dominance solvable* games.

In the rest of this section we consider three versions of the MO-HT game, by focusing on the strategy and payoff of the defender. As we will see, in our setup a dominant strategy exists for the defender, hence opening the way to the derivation of the equilibrium point of the game. Such results are summarized in Theorems 1 through 3 in the sections below. Due to lack of space we report only the proof of Theorem 8, since in our opinion this is the most original proof among the three. The reader may get a feeling about the way the other proofs work by looking at the proof of Theorem 8 or by referring to the proofs of the Theorems in [2].

6.3.2 Notation and definitions

In our framework the system is modeled by a vector of discrete[10] r.v. $\mathbf{X} = X_1, X_2 \dots X_n$ taking values in the same alphabet $\mathcal{X}$. Being related to the same system, the random variables are not independent and hence they are described by means of the joint probability mass function (pmf), say $P_{\mathbf{X}}(x_1, x_2 \dots x_n) = P_{\mathbf{X}}(\mathbf{x})$.

Our analysis relies on the concepts of type and type class defined as follows (see [48] for more details). Given a sequence x^n with elements belonging to an alphabet $\mathcal{X}$, the type P_{x^n} of x^n is the empirical pmf induced by the sequence x^n. In the following, we indicate with $\mathcal{P}_n$ the set of types with denominator n, i.e. the set of types induced by sequences of length n. Given $P \in \mathcal{P}_n$, we indicate with $T(P)$ the type class of P, i.e. the set of all the sequences in $\mathcal{X}^n$ having type P. Being interested in vector sequences, we will also use the vector extension of the above definitions. By considering, for instance, the observation vectors, we indicate by $\mathbf{x}_i = (x_{1,i}, x_{2,i} \dots x_{k,i})$ the vector with the observations of all the nodes at the time instant i, and with $\mathbf{x}^n = \mathbf{x}_1, \mathbf{x}_2 \dots \mathbf{x_n}$ the sequence with all the observed vectors $\mathbf{x}_i$. We then use the notation $P_{\mathbf{x}^n}$ to indicate the empirical joint pmf (the type) induced by the sequence $\mathbf{x}^n$ and with $T(P)$ the type class with all the vector sequences having the empirical pmf equal to P. Finally, we indicate with $\boldsymbol{\mathcal{P}}_n$ all the types for vectors sequence of size k and length n.

Throughout the paper, we adopt a Neyman-Pearson perspective according to which D is interested to accept or reject the hypothesis H_0 that the state is in a safe or normal condition characterized by a pmf $P_{\mathbf{X}}$. In doing so D must ensure that the false positive error probability (P_{fp}) of rejecting H_0 when H_0 holds stays below a threshold. On his side, the attacker aims at inducing a type II error, i.e. to hide the fact that the system exited its normal status. We indicate by $P_{\mathbf{Y}}$ the pmf when H_0 does not hold (H_1). As in [2], we consider an asymptotic version of the problem and require that P_{fp} decays exponentially fast with error exponent at least equal to λ. In addition, we force D to rely on first order statistics only, i.e. to neglect the possible dependence between consecutive observations.

6.3.3 MO-HT with full knowledge

As a first scenario, we consider a simplified case in which the nodes take the observed sequences and pass them to the data fusion center as they are, i.e., $f_l^m = x_l^n, \forall l$. Even if the above condition

[10] Rigorously speaking our analysis is valid only for discrete random variables, the case of continuous variables, however, can be treated by quantizing the continuous alphabet at a resolution which is fine enough.

is rarely verified in practice, this scenario represents a kind of most favorable case for the defender since he can base his decision on all the available information. In addition, the analysis is rather simple since it is a straightforward extension of the game considered in [2]. In the following, we will refer to this scenario as the MO-HT game with full knowledge. Let us, then, define the strategies and payoff of the defender. Mimicking the Neyman-Pearson approach to hypothesis testing, the possible strategies for D are all the possible acceptance regions ensuring a given false positive error probability. In formula:

$$\mathcal{S}_D = \{\Lambda_0 \in 2^{\boldsymbol{\mathcal{P}}_n} \text{ s.t. } P_{fp} \leq 2^{-\lambda n}\}, \tag{31}$$

where Λ_0 is seen as a union of types (a subset of the power set of $\boldsymbol{\mathcal{P}}_n$) due to the limited resources assumption. Thanks to this assumption, in fact, if a vector sequence stays in Λ_0, all the other sequences in the same type class must belong to Λ_0, hence permitting to define Λ_0 as a union of type classes and hence a union of types.

As to the payoff, the defender wishes to minimize the type II error probability, i.e.

$$u_D = -P_{fn} = -\sum_{\mathbf{y}^n : P_{\mathbf{y}^n} \in \Lambda_0} P_{\mathbf{Y}}(\mathbf{y}^n). \tag{32}$$

Our main result regarding the MO-HT game with perfect knowledge is the following.

Theorem 6. *The strategy*

$$\Lambda_0^* = \left\{ P \in \boldsymbol{\mathcal{P}}_n : \mathcal{D}(P||P_{\mathbf{X}}) < \lambda - |\mathcal{X}|^k \frac{\log(n+1)}{n} \right\} \tag{33}$$

where $\mathcal{D}(P||P_{\mathbf{X}})$ indicates the divergence [48] between P and $P_{\mathbf{X}}$, is a dominant strategy for D.

Proof. The proof is virtually identical to the proof of Lemma 1 in [2] and is omitted. □

In practice the fusion center gathers all the observations and verifies if their joint empirical pmf is in accordance with the expected statistics of $\mathbf{X}$ when H_0 holds.

6.3.4 Marginal-based MO-HT

As a second scenario we consider a situation in which the nodes summarize their observations by passing to the fusion center the first order statistics of the observed sequences. In other words, we assume that $m = |\mathcal{X}|$ and $f_l^{|\mathcal{X}|} = P_{x_l^n}$. As an example in which such a scenario applies, we may consider the case of a sensor network in which the nodes observe the system but their link to the fusion center has a very low transmission rate (hypothetically tending to 0). The nodes, then, transmit only the empirical pmf of the observed sequences, i.e. the number of times that each symbol of $\mathcal{X}$ appears in x_l^n. The number of necessary bits to transmit such an information is upper bounded by $|\mathcal{X}| \times \log_2 n$, since each symbol of X may appear in x_l^n at most n times. The rate necessary to transmit this information is hence $\frac{|\mathcal{X}| \times \log_2 n}{n}$, which tends to 0 when $n \to \infty$. Another possible justification for this scenario is the practical difficulty of getting a reliable estimate of the empirical joint pmf. It makes sense, then, for the defender to rely only on the empirical marginal pmf's, but still exploit the knowledge he has on the joint pmf of $\mathbf{X}$.

Given that decision fusion is carried out by considering only the empirical marginal distribution of the vector of observations $\mathbf{x}^n$, the defender is forced to choose a region for H_0 which is a subset of the Cartesian product among the marginal types, i.e. $\mathcal{P}_n^k = \mathcal{P}_n \times \mathcal{P}_n \ldots \mathcal{P}_n$. More precisely we have:

$$\mathcal{S}_D = \{\Lambda_0 \in 2^{\mathcal{P}_n^k} \text{ s.t. } P_{fp} \leq 2^{-\lambda n}\}. \tag{34}$$

As to the payoff, D still aims at minimizing P_{fn} (equation (32)). Finding the optimal acceptance region requires that we compute the probability that a source with a joint pmf $P_{\mathbf{X}}$ emits a sequence having certain marginals. This can be done by considering the probability, under $P_{\mathbf{X}}$, of all the

joint type classes having the desired marginals. To elaborate, let us indicate by $\mathcal{A}_n(P_1, P_2 \dots P_k)$ the set with all joint types with marginals $P_1, P_2 \dots P_k$, that is:

$$\mathcal{A}_n(P_1 \dots P_k) = \{P \in \boldsymbol{\mathcal{P}}_n : \sum_{-i} P(x_1 \dots x_k) = P_i \ \forall i\}, \tag{35}$$

where $\sum_{-i}$ indicates summation over all variables x_j but x_i. Given that the probability of a generic type class Q under $P_{\mathbf{X}}$ decays exponentially fast with exponent $\mathcal{D}(Q||P_{\mathbf{X}})$ and given that the number of types increases polynomially with n, we can proceed as in Lemma 1 in [2] to prove the following theorem.

Theorem 7. *The strategy*

$$\Lambda_0^* = \Bigg\{(P_1 \dots P_k) \in \mathcal{P}_n^k : \min_{P \in \mathcal{A}_n(P_1 \dots P_k)} \mathcal{D}(P||P_{\mathbf{X}}) < \lambda - |\mathcal{X}|^k \frac{\log(n+1)}{n}\Bigg\} \tag{36}$$

is a dominant strategy for D.

Proof. The proof is omitted for sake of brevity. □

One may wonder how the above result changes when the defender does not know $P_{\mathbf{X}}$ but only its marginals. This is the case, for instance, of JPEG forensic tools that analyze separately the DCT coefficients of an image without considering the dependencies between them. In this case it makes sense to adopt a worse case perspective and require that $P_{fp} \leq 2^{-\lambda n}$ for all joint pmf's with assigned marginals. The dominant strategy then includes a double minimization as follows:

$$\Lambda_0^* = \Bigg\{(P_1 \dots P_k) \in \mathcal{P}_n^k : \min_{P_{\mathbf{X}} \in \mathcal{A}(P_{X_1} \dots P_{X_k})} \min_{P \in \mathcal{A}_n(P_1 \dots P_k)} \mathcal{D}(P||P_{\mathbf{X}}) < \lambda - |\mathcal{X}|^k \frac{\log(n+1)}{n}\Bigg\}. \tag{37}$$

6.3.5 MO-HT based on local decisions

The last scenario we considered assumes that the nodes can send to the fusion center only one bit of information. This is a common situation, occurring, for instance but not only, when the nodes take their own decision about the state of the system and data fusion is carried out at the decision level. This scenario also models a multimedia forensic analysis in which the analyst applies several tools each of which provides a binary output regarding the origin or the authenticity of the analyzed document. It is the task of the fusion center to take a final decision by considering the output of all the tools. In principle we would like to derive the optimal decision strategies at the nodes and the optimal fusion strategy. This is a complex task, so we make the simplifying assumption that D adopts an AND fusion strategy, that is H_0 is accepted only if all the nodes accept it. Assuming an AND-based decision rule is equivalent to imposing that the the overall acceptance region is the Cartesian product of the acceptance regions adopted by the nodes, i.e., $\Lambda_0 = \Lambda_{0,1} \times \Lambda_{0,2} \dots \Lambda_{0,k}$. As in the previous sections, we assume that the nodes can rely only on the first order statistics of the observed sequences.

According to the above scenario, the space of strategies of the defender consists of all k-uple of local acceptance regions, that is:

$$\mathcal{S}_D = \{(\Lambda_{0,1} \dots \Lambda_{0,k}) : \Lambda_{0,i} \in 2^{\mathcal{P}_n} \text{ and } P_{fp} \leq 2^{-\lambda n}\}. \tag{38}$$

The payoff function is again the false negative error probability. We now prove the following theorem.

Theorem 8. *The strategy*

$$\Lambda_{0,i}^{*} = \left\{ P_i \in \mathcal{P}_n : \mathcal{D}(P_i||P_{X_i}) < \lambda - |\mathcal{X}|\frac{\log(n+1)}{n} \right\} \ \forall i \tag{39}$$

is a dominant strategy for D.

Proof. The proof consists of two steps. First, we prove that the acceptance region Λ_0^* resulting from the local decision rules defined in (39) is an asymptotically admissible choice for D (i.e. it satisfies the constraint on type I error probability). Then we show that, under the assumption that D adopts an AND fusion rule, the local acceptance regions in (39) minimize the overall type II error probability. Let $\Lambda_{0,i}^{*,c}$ be the rejection region of H_0 at node i. We have:

$$\begin{aligned} P_{fp} &= P_{\mathbf{X}}(\mathbf{x}^n \in \Lambda_0^{*,c}) \\ &= P_{\mathbf{X}}(x_1^n \in \Lambda_{0,1}^{*,c} \text{ OR } x_2^n \in \Lambda_{0,2}^{*,c} \text{ OR } \ldots \text{ OR } x_k^n \in \Lambda_{0,k}^{*,c}) \\ &\leq \sum_{i=1}^{k} P_{X_i}(x_i^n \in \Lambda_{0,i}^{*,c}). \end{aligned} \tag{40}$$

Due to the limited resources assumption, the acceptance region at each node is a union of type classes (or equivalently a union of types with denominator n), hence we can write:

$$\begin{aligned} P_{fp} &\leq \sum_{i=1}^{k} \sum_{P \in \Lambda_{0,i}^{*,c}} P_{X_i}(T(P)) \\ &\overset{a}{\leq} \sum_{i=1}^{k} (n+1)^{|\mathcal{X}|} \max_{P \in \Lambda_{0,i}^{*,c}} P_{X_i}(T(P)) \\ &\overset{b}{\leq} \sum_{i=1}^{k} (n+1)^{|\mathcal{X}|} 2^{-n \min_{P \in \Lambda_{0,i}^{*,c}} \mathcal{D}(P||P_{X_i})} \\ &\overset{c}{\leq} k(n+1)^{|\mathcal{X}|} 2^{-n\left(\lambda - |\mathcal{X}|\frac{\log(n+1)}{n}\right)} \end{aligned} \tag{41}$$

where a and b derive from known upper bound on the number of types with denominator n and on the probability of a type class under a probability measure P_{X_i} [48], and c is a consequence of (39). We have thus shown that $P_{fp} \leq 2^{-n(\lambda - \delta_n)}$ with $\delta_n \to 0$ for $n \to \infty$, and hence Λ_0^* asymptotically satisfies the constraint on P_{fp}.

We now pass to the second part of the proof to show that the strategy in (39) is indeed optimal. Let Λ_0 be an AND-based acceptance region resulting from any other set of local regions $\Lambda_{0,i}$. Let also assume that Λ_0 satisfies the constraint on false positive error probability. Finally, let $\mathbf{x}^{n,*}$ belong to Λ_0^c. This means that $x_i^{n,*} \in \Lambda_{0,i}^c$ for some i, say j. We have:

$$\begin{aligned} 2^{-n\lambda} &\geq P_{\mathbf{X}}(x_i^n \in \Lambda_{0,i}^c, \text{ for some } i) \\ &\overset{a}{\geq} P_{X_j}(x_j^n \in \Lambda_{0,j}^c) \\ &= \sum_{P \in \Lambda_{0,j}^c} P_{X_j}(T(P)) \\ &\overset{b}{\geq} P_{X_j}(T(P_{x_j^{n,*}})) \overset{c}{\geq} \frac{1}{(n+1)^{|\mathcal{X}|}} 2^{-n\left(\mathcal{D}(P_{x_j^{n,*}}||P_{X_j})\right)} \end{aligned} \tag{42}$$

where a is obtained by observing that the probability of a union of events is always larger than the probability of one such events, b holds since we have assumed that $\Lambda_{0,j}^c$ contains at least $x_j^{n,*}$ (and the corresponding type class), and c derives from a known lower bound on the probability of a type class [48]. By considering the first and the last term in (42), we see that $\mathbf{x}^{n,*} \in \Lambda_0^{*,c}$ and hence Λ_0^* is smaller or equal than any other acceptance region satisfying the false positive constraint, thus proving its optimality. □

In practice, according to Theorem 8, H_0 is accepted only if the empirical marginals of the sequences observed by the nodes are in accordance with the system model under H_0. We also note that, somewhat expectedly, D does not exploit the knowledge of the joint pmf $P_{\mathbf{X}}$, the optimum decision rule depending only on the marginal pmf's P_{X_i}.

A unifying, and very important, characteristic of all the scenarios considered in this section, is that the requirement that P_{fp} tends to zero exponentially fast with decay exponent λ and the adoption of a decision rule based on first order statistics already define the optimum defender's strategy regardless of the strategy chosen by attacker, thus resulting in the existence of a dominant strategy for D. Moreover, the dominant strategy does not depend on $P_{\mathbf{Y}}$, that is the statistical characterization of the system when H_0 does not hold, making such a knowledge un-necessary.

6.4 Optimal attacker's strategies

Having derived the optimal strategies for the defender, we now adopt the perspective of the attacker (hereafter referred to as A). The existence of a dominant strategy for D makes it possible to study the optimal attacker's strategy by knowing that the acceptance region adopted by D is equal to Λ_0^*. Together with Λ_0^*, A's optimum strategy defines the equilibrium point of the game, which, being a dominant equilibrium, is also the only rationalizable equilibrium of the game.

6.4.1 Strategy space of the attacker

As a first step, we must define the space of strategies A can choose from and the information he has access to. As detailed in Section 6.2, A acts only when H_0 does not hold with the aim of inducing a type II error. In order to do so, he corrupts either the observation sequences (MO-HT with corrupted observations), or the summaries sent by the nodes to the fusion center (MO-HT with corrupted nodes). In the former case, A must satisfy a distortion constraint specifying to which extent the sequences $x_1^n \dots x_k^n$ can be modified. In both cases, A may be allowed to attack all the sequences or only h of them. In the following, we indicate with y_l^n the observed sequences when H_1 holds and with v_l^m the corresponding feature sequences. The action of the attacker corresponds to applying a function $g(\cdot)$ either to y_l^n or v_l^m to produce k attacked sequences z_l^n (z_l^m in the case of corrupted nodes).

$\mathcal{S}_A$ for MO-HT with corrupted observations. The set of strategies available to A for the MO-HT game with corrupted observations is given by:

$$\mathcal{S}_A = \{g(\cdot) : d(\mathbf{z}^n, \mathbf{y}^n) \leq nD_{max}\}, \tag{43}$$

where D_{max} is the maximum allowed average per letter distortion. Alternatively, we can impose independent constraints on the distortion introduced in each of the observed sequences:

$$\mathcal{S}_A = \{g(\cdot) : d(z_i^n, y_i^n) \leq nD_{i,max} \ \forall i\}. \tag{44}$$

Similar definitions hold when A can corrupt up to h sequences.

$\mathcal{S}_A$ for MO-HT with corrupted nodes. In the case of corrupted nodes the attacker has much more freedom, since in this case he can work directly on the feature sequences v_l^m. All the more that, due to the absence of the distortion constraint, he can replace the feature sequences of the attacked nodes at will. The only applicable constraint is that he can substitute up to h sequences. In the case of chosen corrupted nodes, the space of strategies includes also the choice of the to-be attacked nodes.

Having defined $\mathcal{S}_A$, we must specify the information available to A. To do so, we adopt a worse case assumption and consider an omniscient attacker, who knows the system status (this is implicit in the Neaman-Pearson setup) and can observe all observation and feature sequences, even those that he is not allowed to modify.

6.4.2 Optimum attack for MO-HT with full knowledge

Let us consider the case of corrupted observations first. Given the optimal defender's strategy in (33), it is easy to realize that the optimum strategy for A is to modify the observed sequences so that the divergence between their empirical joint pmf and $P_{\mathbf{X}}$ is as small as possible while satisfying the distortion constraint, that is:

$$g^*(\mathbf{y}^n) = \arg \min_{\mathbf{z}^n : d(\mathbf{z}^n, \mathbf{y}^n) \leq n D_{max}} \mathcal{D}(P_{\mathbf{z}^n} || P_{\mathbf{X}}). \tag{45}$$

This result is analogous to Theorem 1 in [2] (see equation (16) therein), the only difference being that vector sources are involved instead of scalar ones. A similar result holds when the distortion constraint applies to each observed sequence separately. Note that, even if theoretically simple, solving the minimization in (45) may be computationally very expensive, as already pointed out in [2] for the scalar case. In the case of MO-HT with corrupted nodes, the situation is by far more favorable to the attacker, since he has to solve the minimization problem without any constraint. It is obvious, then, that A can pass to the fusion center completely fake sequences for which the divergence between the empirical joint pmf and $P_{\mathbf{X}}$ is arbitrarily small, thus always resulting in a false negative error.

The situation is different when A can attack only h out of k nodes. Even in the most favorable case of corrupted nodes, A can not control the empirical marginals of the non-attacked nodes and the joint pmf between them. If such marginals, or joint pmf, under H_1 are different from those under H_0, it may still be possible for the defender to reliably distinguish between the two hypothesis (though with a higher P_{fn}). It is also evident, that in the case of chosen corrupted nodes, A will attack the nodes for which the pmf's of the observations under H_0 and H_1 differ most.

6.4.3 Optimum attack for Marginal-based MO-HT

Even in this case the optimal attacking strategy follows directly from the knowledge of D's dominant strategy. In fact, for the case of corrupted observable, from equation (36), it follows that:

$$g^*(\mathbf{y}^n) = \arg \min_{\mathbf{z}^n : d(\mathbf{z}^n, \mathbf{y}^n) \leq n D_{max}} \min_{P \in \mathcal{A}_n(P_{z_1^n} \dots P_{z_k^n})} \mathcal{D}(P || P_{\mathbf{X}}). \tag{46}$$

A similar result holds when equation (37) applies instead of (36). The situation is more favorable when the attacker can corrupt the output of the nodes, since in this case he can choose directly the $P_{z_1^n} \dots P_{z_k^n}$ that minimize $\min_{P \in \mathcal{A}_n(P_{z_1^n} \dots P_{z_k^n})} \mathcal{D}(P||P_{\mathbf{X}})$. In fact, by letting $z_i^m = P_{z_i^n} = P_{X_i}$ for all i, we have a perfect attack, since in this case $\min_{P \in \mathcal{A}_n(P_{z_1^n} \dots P_{z_k^n})} \mathcal{D}(P||P_{\mathbf{X}})$ is obviously equal to 0. Of course, this is not possible when the attacker controls only h nodes, in which case the optimum attack boils down to the following minimization[11]:

$$\min_{z_1^m \dots z_h^m} \min_{P \in \mathcal{A}_n(z_1^m, \dots z_h^m, P_{y_{h+1}^n} \dots P_{y_k^n})} \mathcal{D}(P || P_{\mathbf{X}}). \tag{47}$$

Finally, when the attacker chooses which nodes to attack, a further minimization is required to minimize (47) over all possible subsets of attacked nodes.

6.4.4 Optimum attack for MO-HT based on local decisions

Once again the optimum attacker's strategy follows directly from the knowledge of the dominant strategy of the defender. By considering Theorem 8, in fact, is easy to conclude that the optimum strategy for A in the case of corrupted observations is:

$$g^*(\mathbf{y}^n) = \arg \min_{\mathbf{z}^n : d(\mathbf{z}^n, \mathbf{y}^n) \leq n D_{max}} \min_i \mathcal{D}(P_{z_i^n} || P_{X_i}). \tag{48}$$

[11] We assume w.l.o.g. that A attacks the first h nodes.

As before the derivation of the optimum attack may be computationally expensive due to the presence of the distance constraint. If the squared Euclidean distance is adopted, a kind of waterfilling approach can be applied. The attacker, in fact, can operate as follows: choose i such that $\mathcal{D}(P_{y_i^n}||P_{X_i})$ is maximum, and compute z_i^n such that $\mathcal{D}(P_{z_i^n}||P_{X_i}) = \lambda - |\mathcal{X}|\log(n+1)/n - \varepsilon$ (with ε arbitrarily small), and the squared Euclidean distance between z_i^n and y_i^n is minimum. If the distortion is lower than nD_{max}, go on with the next i such that $\mathcal{D}(P_{y_i^n}||P_{X_i})$ is maximum, and iterate the above procedure until all $\mathcal{D}(P_{y_i^n}||P_{X_i})$ are lower than the decision threshold or when the maximum distortion is reached.

A considerably simpler situation is obtained when separate distortion constraints apply to the different sequences. In this case in fact, the attacker has to solve k independent scalar minimizations.

To conclude, we consider the case of corrupted nodes. In this case the optimum attack is trivial, since the attacker needs only to set the output of all the nodes under his control to 0. Note however, that this may not be enough if A does not control all the nodes, since the fusion center accepts H_0 only if all the nodes accept it.

In the case of chosen attacked nodes, A will attack the nodes for which the marginals under H_1 differ most from those under H_0.

6.5 Discussion and conclusions

We conclude this section by drawing some conclusions and summarizing the main lessons that we learnt from our analysis.

To start with, we observe that the theoretical framework with the taxonomy of several kinds of scenarios referring to different practical applications, is by itself a fundamental step towards the comprehension of the addressed problems and the development of practical strategies for both the attacker and the defender.

With regard to the specific results we have proven, the most striking result regards the existence of a dominant strategy for the defender. What Theorems 1 through 3 say, in fact, is that the defender may chose its strategy without caring about the attacker. For instance, he would get no advantage from the knowledge of the attacked nodes, let alone from any attempt to discover them. This marks an important difference with respect to previous works in which the defender tries to distinguish between honest and malicious nodes. In hindsight, the reason for such an apparently strange behavior, is the adoption of a Neyman-Pearson setup wherein the attacker acts only when H_0 does not hold, while the defender is asked to satisfy a requirement on P_{fp}, i.e., by assuming that H_0 holds. Coupled with the adoption of an asymptotic setup, this results in the existence of a dominant strategy for D that does not need to know whether a node (or an observation) is controlled by the adversary or not. It goes without saying that is some applications the assumptions we made may not be reasonable, thus opening the way to different formulations of the MO-HT game.

Having determined the equilibrium point of the game, the next step would require that the payoff at the equilibrium is evaluated so to know who is going to *win* the game. In other words, given the pmf's under H_0 and H_1 (res. $P_{\mathbf{X}}$ and $P_{\mathbf{Y}}$), and a distortion constraint D_{max} (when applicable), we would like to know whether the probability of a type II error ultimately tends to 0 or 1 when $n \to \infty$. Doing so for $\lambda \to 0$ would finally permit us to decide whether the two hypothesis H_0 and H_1 are ultimately distinguishable or not, when the attacks is allowed to attack h observation sequences (or nodes) with a maximum per letter distortion D_{max}.

TODO: Check coherence between references and remove duplicates

References

[1] M. Barni and F. Perez-Gonzalez, "Coping with the enemy: Advances in adversary-aware signal processing," in *ICASSP 2013, IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, CA, May 2013, pp. 8682–8686. 4, 58

[2] M. Barni and B. Tondi, "The source identification game: an information-theoretic perspective," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 3, pp. 450–463, March 2013. 4, 58, 61, 62, 63, 66

[3] A. P. Dempster, "Upper and lower probabilities induced by a multivalued mapping," *Annals of Mathematical Statistics*, vol. 38, pp. 325–339, 1967. 8, 43

[4] G. Shafer, *A Mathematical Theory of Evidence*. Princeton: Princeton University Press, 1976. 8

[5] R. Yager, "Aggregating non-independent Dempster-Shafer belief structures," in *IPMU 2008, International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Malaga, ES, June 2008, pp. 289–297. 9

[6] A. Benavoli, L. Chisci, B. Ristic, A. Farina, and A. Graziano, *Reasoning under uncertainty: from Bayesian to Valuation Based Systems. Application to target classification and threat evaluation.* Rome, ITA: SELEX Sistemi Integrati, 2007. 10

[7] L. A. Zadeh, "Review of a mathematical theory of evidence," *AI magazine*, vol. 5, no. 3, p. 81, 1984. 10

[8] A. C. Doyle, *The sign of four*. Lippincott's Monthly Magazine, 1980. 11

[9] T. Bianchi, A. De Rosa, and A. Piva, "Improved DCT coefficient analysis for forgery localization in JPEG images," in *ICASSP 2011, IEEE International Conference on Acoustics, Speech and Signal Processing*, Prague, CZ, May 2011, pp. 2444 – 2447. 18, 19, 26, 42, 43, 44, 45, 50

[10] M. Barni and A. Costanzo, "A fuzzy approach to deal with uncertainty in image forensics," *Signal Processing: Image Communication*, vol. 27, no. 9, pp. 998–1010, 2012. 19

[11] M. Fontani, T. Bianchi, A. De Rosa, A. Piva, and M. Barni, "A Dempster-Shafer framework for decision fusion in image forensics," in *WIFS 2011, IEEE International Workshop on Information Forensics and Security*, Foz do Iguaçu, BR, December 2011, pp. 1–6. 19

[12] M. Fontani, T. Bianchi, A. De Rosa, A. Piva, and M. Barni, "A Framework for Decision Fusion in Image Forensics Based on Dempster-Shafer Theory of Evidence," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 4, pp. 593–607, 2013. 19

[13] T. Denoeux, "A k-nearest neighbor classification rule based on Dempster-Shafer theory," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 25, no. 5, pp. 804–813, 1995. 20

[14] L. Ceriani and P. Verme, "The origins of the Gini index: extracts from Variabilità e Mutabilità (1912) by Corrado Gini," *Journal of Economic Inequality*, vol. 10, no. 3, pp. 421–443, September 2012. 22

[15] S. Bayram, B. Sankur, N. Memon, and İ. Avcıbaş, "Image manipulation detection," *Journal of Electronic Imaging*, vol. 15, no. 4, pp. 041 102–041 102, 2006. 24

[16] M. Kharrazi, H. T. Sencar, and N. Memon, "Improving steganalysis by fusion techniques: A case study with image steganography," *Transactions on Data Hiding and Multimedia Security I*, pp. 123–137, 2006. 24

[17] Y.-F. Hsu and S.-F. Chang, "Statistical fusion of multiple cues for image tampering detection," in *Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, October 2008, pp. 1386–1390. 24

[18] G. Chetty and M. Singh, "Nonintrusive image tamper detection based on fuzzy fusion," *International Journal of Computer Science and Network Security*, vol. 10, no. 9, pp. 86–90, September 2010. 24

[19] D. Hu, L. Wang, Y. Zhou, Y. Zhou, X. Jiang, and L. Ma, "Ds evidence theory based digital image trustworthiness evaluation model," in *MINES 2009, International Conference on Multimedia Information Networking and Security*, vol. 1. Hubei, CHN: IEEE, November 2009, pp. 85–89. 24

[20] P. Zhang and X. Kong, "Detecting image tampering using feature fusion," in *ARES 2009, International Conference on Availability, Reliability and Security.* Fukuoka, JP: IEEE, March 2009, pp. 335–340. 24

[21] W. Luo, Z. Qu, J. Huang, and G. Qiu, "A novel method for detecting cropped and recompressed image block," in *ICASSP 2007, IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, Honolulu, USA, April 2007, pp. II–217 –II–220. 26, 50

[22] T. Bianchi and A. Piva, "Detection of non-aligned double JPEG compression with estimation of primary compression parameters," in *ICIP 2011, IEEE International Conference on Image Processing*, Brussels, BE, September 2011, pp. 1929 –1932. 26, 27, 42, 45, 50

[23] Z. C. Lin, J. F. He, X. Tang, and C. K. Tang, "Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis," *Pattern Recognition*, vol. 42, no. 11, pp. 2492–2501, November 2009. 26, 42, 44, 50

[24] H. Farid, "Exposing digital forgeries from JPEG ghosts," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 1, pp. 154–160, 2009. 26, 27, 38, 50, 57

[25] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm. 26

[26] M. Barni, A. Costanzo, and L. Sabatini, "Identification of cut & paste tampering by means of double-JPEG detection and image segmentation," in *ISCAS 2010, IEEE International Symposium on Circuits and Systems*, Paris, FR, May 2010, pp. 1687–1690. 41

[27] M. Chen, J. Fridrich, M. Goljan, and J. Lukas, "Determining image origin and integrity using sensor noise," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 1, pp. 74–90, 2008. 41

[28] A. Swaminathan, M. Wu, and K. J. R. Liu, "Digital image forensics via intrinsic fingerprints," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 1, pp. 101–117, 2008. 41

[29] P. Ferrara, T. Bianchi, A. De Rosa, and A. Piva, "Image forgery localization via fine-grained analysis of CFA artifacts," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 5, pp. 1566–1577, October 2012. 42, 44, 45

[30] G. Chierchia, D. Cozzolino, G. Poggi, C. Sansone, and L. Verdoliva, "Guided filtering for PRNU-based localization of small-size image forgeries," in *ICASSP 2014, IEEE International Conference on Acoustics, Speech and Signal Processing*, Florence, ITA, May 2014, pp. 6231 – 6235. 45

[31] K. He, J. Sun, and X. Tang, "Guided image filtering," in *ECCV 2010, European Conference on Computer Vision*, ser. Lecture Notes in Computer Science, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Springer Berlin Heidelberg, 2010, vol. 6311, pp. 1–14. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-15549-9_1 45

[32] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama, "Digital photography with flash and no-flash image pairs," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 664–672, Aug. 2004. [Online]. Available: http://doi.acm.org/10.1145/1015706.1015777 45

[33] M. Fontani, E. Argones-Rúa, C. Troncoso, and M. Barni, "The Watchful Forensic Analyst: Multi-Clue Information Fusion with Background Knowledge," in *WIFS 2013, IEEE International Workshop on Information Forensics and Security*, November 2013, pp. 1–6. 45, 46, 57

[34] D. Cozzolino, F. Gargiulo, C. Sansone, and L. Verdoliva, "Multiple classifier systems for image forgery detection," in *Image Analysis and Processing*. Springer, 2013, vol. 8157, pp. 259–268. 47

[35] R. Böhme and M. Kirchner, "Counter-forensics: Attacking image forensics," in *Digital Image Forensics*, H. T. Sencar and N. Memon, Eds. Springer New York, 2013, pp. 327–366. [Online]. Available: http://dx.doi.org/10.1007/978-1-4614-0757-7_12 48

[36] M. Barni, M. Fontani, and B. Tondi, "A universal attack against histogram-based image forensics," *International Journal of Digital Crime and Forensics*, vol. 5, no. 3, pp. 35–52, 2013. 48

[37] M. Kirchner and R. Bohme, "Hiding traces of resampling in digital images," *Information Forensics and Security, IEEE Transactions on*, vol. 3, no. 4, pp. 582–592, 2008. 49

[38] M. Kirchner and J. Fridrich, "On detection of median filtering in digital images," in *IS&T/SPIE Electronic Imaging 2010*, ser. SPIE Conference Series, vol. 7541, San Jose, USA, January 2010, pp. 754 110–754 110. 51

[39] T. Pevny, P. Bas, and J. Fridrich, "Steganalysis by subtractive pixel adjacency matrix," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 215–224, June 2010. 51

[40] M. Stamm and K. Liu, "Forensic detection of image manipulation using statistical intrinsic fingerprints," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 3, pp. 492–506, September 2010. 55, 57

[41] G. Valenzise, M. Tagliasacchi, and S. Tubaro, "Revealing the traces of JPEG compression anti-forensics," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 2, pp. 335–349, 2013. 55, 57

[42] M. Barni and B. Tondi, "Binary hypothesis testing game with training data," *IEEE Transactions on Information Theory*, vol. 60, no. 8, pp. 4848–4866, August 2014. 58

[43] P. K. Varshney, *Distributed Detection and Data Fusion*. Springer-Verlag, 1997. 59

[44] J.-F. Chamberland and V. V. Veeravalli, "Decentralized detection in sensor networks," *IEEE Transactions on Signal Processing*, vol. 51, no. 2, pp. 407–416, February 2003. 59

[45] I. F. Akyildiz, B. F. Lo, and R. Balakrishnan, "Cooperative spectrum sensing in cognitive radio networks: A survey," *Physical Communication*, vol. 2011, no. 4, pp. 40–62, 2011. 59

[46] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*. MIT Press, 1994. 60

[47] Y. C. Chen, N. Van Long, and X. Luo, "Iterated strict dominance in general games," *Games and Economic Behavior*, vol. 61, no. 2, pp. 299–315, November 2007. 61

[48] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley Interscience, 1991. 61, 62, 64